Fei Liu
Nan Duan
Qingting Xu
Yu Hong (Eds.)

LNAI 14302

Natural Language Processing and Chinese Computing

12th National CCF Conference, NLPCC 2023
Foshan, China, October 12–15, 2023
Proceedings, Part I

1 Part I

中国计算机学会
CCF

Springer

MOREMEDIA

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence 14302

The series Lecture Notes in Artificial Intelligence (LNAI) was established in 1988 as a topical subseries of LNCS devoted to artificial intelligence.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Fei Liu · Nan Duan · Qingting Xu · Yu Hong
Editors

Natural Language Processing and Chinese Computing

12th National CCF Conference, NLPCC 2023
Foshan, China, October 12–15, 2023
Proceedings, Part I

Springer

Editors

Fei Liu
Emory University
Atlanta, GA, USA

Nan Duan
Microsoft Research Asia
Beijing, China

Qingting Xu
Soochow University
Suzhou, China

Yu Hong
Soochow University
Suzhou, China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-44692-4 ISBN 978-3-031-44693-1 (eBook)
https://doi.org/10.1007/978-3-031-44693-1

LNCS Sublibrary: SL7 – Artificial Intelligence

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

Welcome to NLPCC 2023, the twelfth CCF International Conference on Natural Language Processing and Chinese Computing. Following the success of previous conferences held in Beijing (2012), Chongqing (2013), Shenzhen (2014), Nanchang (2015), Kunming (2016), Dalian (2017), Hohhot (2018), Dunhuang (2019), Zhengzhou (2020), Qingdao (2021), and Guilin (2022), this year's NLPCC will be held in Foshan. As a premier international conference on natural language processing and Chinese computing, organized by the CCF-NLP (Technical Committee of Natural Language Processing, China Computer Federation, formerly known as Technical Committee of Chinese Information, China Computer Federation), NLPCC serves as an important forum for researchers and practitioners from academia, industry, and government to share their ideas, research results, and experiences, and to promote their research and technical innovations.

The fields of natural language processing (NLP) and Chinese computing (CC) have boomed in recent years. Following NLPCC's tradition, we welcomed submissions in ten areas for the main conference: Fundamentals of NLP; Machine Translation and Multilinguality; Machine Learning for NLP; Information Extraction and Knowledge Graph; Summarization and Generation; Question Answering; Dialogue Systems; Large Language Models; NLP Applications and Text Mining; Multimodality and Explainability. This year, we received 478 valid submissions to the main conference on the submission deadline.

After a thorough reviewing process, including meta reviewing, out of 478 valid submissions (some of which were withdrawn by authors or desk-rejected due to policy violations), 134 papers were finally accepted as regular papers to appear in the main conference, resulting in an acceptance rate of 29.9%. Among them, 64 submissions will be presented as oral papers and 79 as poster papers at the conference. 5 papers were nominated by our area chairs for the best paper award. An independent best paper award committee was formed to select the best papers from the shortlist. This proceeding includes only the accepted English papers; the Chinese papers will appear in the ACTA Scientiarum Naturalium Universitatis Pekinensis. In addition to the main proceedings, 3 papers were accepted to the Student workshop, 32 papers were accepted to the Evaluation workshop.

We are honored to have four internationally renowned keynote speakers, Denny Zhou (Google Deepmind), Xia (Ben) Hu (Rice University), Arman Cohan (Yale University), and Diyi Yang (Stanford University), sharing their findings on recent research progress and achievements in natural language processing.

We would like to thank all the people who have contributed to NLPCC 2023. First of all, we would like to thank our 21 area chairs for their hard work recruiting reviewers, monitoring the review and discussion processes, and carefully rating and recommending submissions. We would like to thank all 322 reviewers for their time and efforts to review the submissions. We are also grateful for the help and support from the general chairs, Rada Mihalcea and Hang Li, and from the organization committee chairs, Biqin Zeng, Yi Cai and Xiaojun Wan. Special thanks go to Yu Hong and Qingting Xu, the publication chairs. We greatly appreciate all your help!

Finally, we would like to thank all the authors who submitted their work to NLPCC 2023, and thank our sponsors for their contributions to the conference. Without your support, we could not have such a strong conference program.

We are happy to see you at NLPCC 2023 in Foshan and hope you enjoy the conference!

August 2023

Fei Liu
Nan Duan

Organization

NLPCC 2023 is organized by China Computer Federation (CCF), and hosted by South China Normal University. Publishers comprise Lecture Notes on Artificial Intelligence (LNAI), Springer Verlag, and ACTA Scientiarum Naturalium Universitatis Pekinensis.

Organization Committee

General Chairs

Rada Mihalcea	University of Michigan
Hang Li	ByteDance Technology

Program Committee Chairs

Fei Liu	Emory University
Nan Duan	Microsoft Research Asia

Student Workshop Chairs

Jing Li	The Hong Kong Polytechnic University
Jingjing Wang	Soochow University

Evaluation Chairs

Yunbo Cao	Tencent
Piji Li	Nanjing University of Aeronautics and Astronautics

Tutorial Chairs

Zhongyu Wei	Fudan University
Zhaochun Ren	Shandong University

Publication Chairs

Yu Hong	Soochow University
Qingting Xu	Soochow University

Journal Coordinator

Yunfang Wu	Peking University

Conference Handbook Chair

Leixin Du	South China Normal University

Sponsorship Chairs

Min Zhang	Harbin Institute of Technology (Shenzhen)
Haofen Wang	Tongji University
Ruifeng Xu	Harbin Institute of Technology (Shenzhen)

Publicity Chairs

Benyou Wang	The Chinese University of Hong Kong (Shenzhen)
Shen Gao	Shandong University
Xianling Mao	Beijing Institute of Technology

Organization Committee Chairs

Biqin Zeng	South China Normal University
Yi Cai	South China University of Technology
Xiaojun Wan	Peking University

Treasurer

Yajing Zhang	Soochow University
Xueying Zhang	Peking University

Webmaster

Hui Liu	Peking University

Program Committee

Xiang Ao	Institute of Computing Technology, Chinese Academy of Sciences, China
Jiaxin Bai	Hong Kong University of Science and Technology, China
Xinyi Bai	Google, USA
Junwei Bao	JD AI Research, China
Qiming Bao	The University of Auckland, New Zealand
Xiangrui Cai	Nankai University, China
Shuyang Cao	Univerisity of Michigan, USA
Zhangming Chan	Alibaba Group, China
Yufeng Chen	Beijing Jiaotong University, China
Yulong Chen	Zhejiang University, Westlake University, China
Bo Chen	Minzu University of China, China
Jianghao Chen	CASIA, China
Wenhu Chen	University of Waterloo & Google Research, Canada
Yubo Chen	Institute of Automation, Chinese Academy of Sciences, China
Xuelu Chen	UCLA, USA
Yidong Chen	Department of Artificial Intelligence, School of Informatics, Xiamen University, China
Chen Chen	Nankai University, China
Guanyi Chen	Utrecht University, Netherlands
Qi Chen	Northeastern University, China
Wenliang Chen	Soochow University, China
Xinchi Chen	Amazon AWS, USA
Muhao Chen	USC, USA
Liang Chen	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Jiangjie Chen	Fudan University, China
Leshang Chen	Oracle America, Inc., USA
Sihao Chen	University of Pennsylvania, USA
Wei Chen	School of Data Science, Fudan University, China
Kewei Cheng	UCLA, USA
Cunli Mao	Kunming University of Science and Technology, China

Xiang Deng	The Ohio State University, USA
Chenchen Ding	NICT, Japan
Qianqian Dong	ByteDance AI Lab, China
Yue Dong	University of California Riverside, USA
Zi-Yi Dou	UCLA, USA
Rotem Dror	University of Pennsylvania, Israel
Xinya Du	University of Texas at Dallas, USA
Junwen Duan	Central South University, China
Chaoqun Duan	JD AI Research, China
Nan Duan	Microsoft Research Asia, China
Xiangyu Duan	Soochow University, China
Alex Fabbri	Salesforce AI Research, USA
Zhihao Fan	Fudan University, China
Tianqing Fang	Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Zichu Fei	Fudan University, China
Shi Feng	Northeastern University, China
Jiazhan Feng	Peking University, China
Yang Feng	Institute of Computing Technology, Chinese Academy of Sciences, China
Zhangyin Feng	Harbin Institute of Technology, China
Yansong Feng	Peking University, China
Xingyu Fu	Upenn, USA
Guohong Fu	Soochow University, China
Yi Fung	University of Illinois at Urbana Champaign, USA
Shen Gao	Shandong University, China
Heng Gong	Harbin Institute of Technology, China
Yeyun Gong	Microsoft Research Asia, China
Yu Gu	The Ohio State University, USA
Yi Guan	School of Computer Science and Technology, Harbin Institute of Technology, China
Tao Gui	Fudan University, China
Daya Guo	Sun Yat-Sen University, China
Shaoru Guo	Institute of Automation, Chinese Academy of Sciences, China
Yiduo Guo	Peking University, China
Jiale Han	Beijing University of Posts and Telecommunications, China
Xudong Han	The University of Melbourne, Australia
Lifeng Han	The University of Manchester, UK
Xianpei Han	Institute of Software, Chinese Academy of Sciences, China

Tianyong Hao	School of Computer Science, South China Normal University, China
Hongkun Hao	Shanghai Jiao Tong University, China
Ruifang He	Tianjin University, China
Xudong Hong	Saarland University/MPI Informatics, Germany
I-Hung Hsu	USC Information Sciences Institute, USA
Zhe Hu	Baidu, China
Junjie Hu	University of Wisconsin-Madison, USA
Xiaodan Hu	University of Illinois at Urbana-Champaign, USA
Minghao Hu	Information Research Center of Military Science, China
Shujian Huang	National Key Laboratory for Novel Software Technology, Nanjing University, China
Fei Huang	Tsinghua University, China
Baizhou Huang	Peking University, China
Junjie Huang	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Yueshan Huang	University of California San Diego, China
Qingbao Huang	Guangxi University, China
Xin Huang	Institute of Automation, Chinese Academy of Sciences, China
James Y. Huang	University of Southern California, USA
Jiangping Huang	Chongqing University of Posts and Telecommunications, China
Changzhen Ji	Hithink RoyalFlush Information Network, China
Chen Jia	Fudan University, China
Tong Jia	Northeastern University, China
Hao Jia	School of Computer Science and Technology, Soochow University, China
Hao Jiang	University of Science and Technology of China, China
Wenbin Jiang	Baidu Inc., China
Jingchi Jiang	Harbin Institute of Technology, China
Huiming Jin	Apple Inc., USA
Feihu Jin	Institute of Automation, Chinese Academy of Sciences, China
Peng Jin	Leshan Normal University, China
Zhu Junguo	Kunming University of Science and Technology, China
Lingpeng Kong	The University of Hong Kong, Hong Kong Special Administrative Region of China
Fajri Koto	MBZUAI, United Arab Emirates

Tuan Lai	University of Illinois at Urbana-Champaign, USA
Yuxuan Lai	Peking University, China
Yuanyuan Lei	Texas A&M University, USA
Maoxi Li	School of Computer Information Engineering, Jiangxi Normal University, China
Zekun Li	University of Minnesota, USA
Bei Li	Northeastern University, China
Chenliang Li	Wuhan University, China
Piji Li	Nanjing University of Aeronautics and Astronautics, China
Zejun Li	Fudan University, China
Yucheng Li	University of Surrey, UK
Yanran Li	The Hong Kong Polytechnic University, China
Shasha Li	College of Computer, National University of Defense Technology, China
Mingda Li	University of California, Los Angeles, USA
Dongfang Li	Harbin Institute of Technology, Shenzhen, China
Zuchao Li	Wuhan University, China
Mingzhe Li	Peking University, China
Miao Li	The University of Melbourne, Australia
Jiaqi Li	iFlytek Research (Beijing), China
Chenxi Li	UCSD, USA
Yanyang Li	The Chinese University of Hong Kong, China
Fei Li	Wuhan University, China
Jiajun Li	Shanghai Huawei Technology Co., Ltd., China
Jing Li	Department of Computing, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region of China
Fenghuan Li	Guangdong University of Technology, China
Zhenghua Li	Soochow University, China
Qintong Li	The University of Hong Kong, Hong Kong Special Administrative Region of China
Haonan Li	MBZUAI, United Arab Emirates
Zheng Li	Stockton University, USA
Bin Li	Nanjing Normal University, China
Yupu Liang	University of Chinese Academy of Sciences, China
Yaobo Liang	Microsoft, China
Lizi Liao	Singapore Management University, Singapore
Ye Lin	Northeastern University, China
Haitao Lin	National Laboratory of Pattern Recognition, Institute of Automation, CAS, China
Jian Liu	Beijing Jiaotong University, China

Xianggen Liu	Sichuan University, China
Fei Liu	Emory University, USA
Yuanxing Liu	Harbin Institute of Technology, China
Xuebo Liu	Harbin Institute of Technology, Shenzhen, China
Pengyuan Liu	Beijing Language and Culture University, China
Lemao Liu	Tencent AI Lab, China
Chunhua Liu	The University of Melbourne, Australia
Qin Liu	University of Southern California, USA
Tianyang Liu	University of California San Diego, USA
Puyuan Liu	University of Alberta, Canada
Qian Liu	Sea AI Lab, Singapore
Shujie Liu	Microsoft Research Asia, Beijing, China, China
Kang Liu	Institute of Automation, Chinese Academy of Sciences, China
Yongbin Liu	School of Computer Science, University of South China, China
Zhenhua Liu	School of Computer Science and Technology, Soochow University, China, China
Xiao Liu	Microsoft Research Asia, China
Qun Liu	Chongqing University of Posts and Telecommunications, China
Yunfei Long	University of Essex, UK
Renze Lou	Pennsylvania State University, USA
Keming Lu	University of Southern California, USA
Jinliang Lu	National Laboratory of Pattern Recognition, CASIA, Beijing, China, China
Jinzhu Lu	Fudan University, China
Xin Lu	Harbin Institute of Technology, China
Shuai Lu	Microsoft, China
Hengtong Lu	Beijing University of Posts and Telecommunications, China
Minghua Ma	Microsoft, China
Cong Ma	Institute of Automation, Chinese Academy of Sciences; University of Chinese Academy of Sciences, China
Mingyu Derek Ma	UCLA, USA
Yinglong Ma	North China Electric Power University, China
Yunshan Ma	National University of Singapore, Singapore
Xianling Mao	Beijing Institute of Technology, China
Zhao Meng	ETH Zurich, Switzerland
Xiangyang Mou	Meta, USA
Minheng Ni	Microsoft Research, China
Yasumasa Onoe	The University of Texas at Austin, USA

Zhufeng Pan	Google, USA
Xutan Peng	Huawei, China
Ehsan Qasemi	University of Southern California, USA
Weizhen Qi	University of Science and Technology of China, China
Tao Qian	Hubei University of Science and Technology, China
Yanxia Qin	School of Computing, National University of Singapore, Singapore
Zixuan Ren	Institute of Automation, China
Stephanie Schoch	University of Virginia, USA
Lei Sha	Beihang University, China
Wei Shao	City University of Hong Kong, China
Zhihong Shao	Tsinghua University, China
Haoran Shi	Amazon Inc., USA
Xing Shi	Bytedance Inc., China
Jyotika Singh	Placemakr, USA
Kaiqiang Song	Tencent AI Lab, USA
Haoyu Song	Harbin Institute of Technology, China
Zhenqiao Song	UCSB, China
Jinsong Su	Xiamen University, China
Dianbo Sui	Harbin Institute of Technology, China
Zequn Sun	Nanjing University, China
Kexuan Sun	University of Southern California, USA
Chengjie Sun	Harbin Institute of Technology, China
Kai Sun	Meta, USA
Chuanyuan Tan	Soochow University, China
Zhixing Tan	Zhongguancun Laboratory, China
Minghuan Tan	Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
Ping Tan	Universiti Malaysia Sarawak, Malaysia
Buzhou Tang	Harbin Institute of Technology (Shenzhen), China
Rongchuan Tang	Institute of Automation, Chinese Academy of Sciences, China
Xiangru Tang	Yale University, USA
Duyu Tang	Tencent, China
Xunzhu Tang	University of Luxembourg, Luxembourg
Mingxu Tao	Peking University, China
Zhiyang Teng	Nanyang Technological University, Singapore
Xiaojun Wan	Peking University, China
Chen Wang	National Laboratory of Pattern Recognition, Institute of Automation, CAS, China

Lijie Wang	Baidu, China
Liang Wang	Microsoft Research, China
Xinyuan Wang	University of California, SanDiego, USA
Haoyu Wang	University of Pennsylvania, USA
Xuesong Wang	Harbin Institute of Technology, China
Hongwei Wang	Tencent AI Lab, USA
Hongling Wang	Soochow University, China
Ke Wang	Huawei Technologies Ltd., China
Qingyun Wang	University of Illinois at Urbana-Champaign, USA
Yiwei Wang	Amazon, USA
Jun Wang	University of Melbourne, Australia
Jingjing Wang	Soochow University, China
Ruize Wang	Academy for Engineering and Technology, Fudan University, China
Zhen Wang	The Ohio State University, USA
Qiang Wang	Hithink RoyalFlush AI Research Institute, China, China
Lingzhi Wang	The Chinese University of Hong Kong, China
Yufei Wang	Macquaire University, Australia
Xun Wang	Microsoft, USA
Sijia Wang	Virginia Tech, USA
Yaqiang Wang	Chengdu University of Information Technology, China
Siyuan Wang	Fudan University, China
Xing Wang	Tencent, China
Fei Wang	University of Southern California, USA
Gengyu Wang	Columbia University, USA
Tao Wang	Department of Biostatistics & Health Informatics, King’s College London, UK
Bo Wang	Tianjin University, China
Wei Wei	Huazhong University of Science and Technology, China
Bingbing Wen	The University of Washington, USA
Lianwei Wu	School of Software Engineering, Xi’an Jiaotong University, China
Chenfei Wu	Microsoft, China
Ting Wu	Fudan University, China
Yuxia Wu	Xi’an Jiaotong University, China
Sixing Wu	School of Software, Yunnan University, China
Junhong Wu	Cognitive Computing Lab, Peking University, China
Shuangzhi Wu	Bytedance, China
Lijun Wu	Microsoft Research, China

Tengxiao Xi	University of Chinese Academy of Sciences School of Artificial Intelligence, China
Yang Xiang	Peng Cheng Laboratory, China
Tong Xiao	Northeastern University, China
Min Xiao	State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, CAS, China
Ye Xiao	Hunan University, China
Jun Xie	Alibaba DAMO Academy, China
Yuqiang Xie	Institute of Information Engineering, Chinese Academy of Sciences, China
Qingting Xu	Soochow University, China
Jinan Xu	Beijing Jiaotong University, China
Yiheng Xu	Microsoft Research Asia, China
Chen Xu	Northeastern University, China
Kang xu	Nanjing University of Posts and Telecommunications, China
Wang Xu	Harbin Institute of Technology, China
Jiahao Xu	Nanyang Technological University, Singapore
Nan Xu	University of Southern California, USA
Yan Xu	Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Hanzi Xu	Temple University, USA
Zhixing Xu	Nanjing Normal University, China
Jiacheng Xu	Salesforce AI Research, USA
Xiao Xu	Harbin Institute of Technology, China
Rui Yan	Renmin University of China, China
Kun Yan	Beihang University, China
Lingyong Yan	Baidu Inc., China
Baosong Yang	Alibaba Damo Academy, Alibaba Inc., China
Shiquan Yang	The University of Melbourne, Australia
Haoran Yang	The Chinese University of Hong Kong, Hong Kong Special Administrative Region of China
Liang Yang	Dalian University of Technology, China
Jun Yang	Marcpoint Co., Ltd., China
Muyun Yang	Harbin Institute of Technology, China
Kai Yang	Zhongguancun Laboratory, China
Zhiwei Yang	College of Computer Science and Technology, Jilin University, China
Ziqing Yang	CISPA Helmholtz Center for Information Security, Germany

Jianmin Yao	Soochow University, China
Shengming Yin	University of Science and Technology of China, China
Wenpeng Yin	Pennsylvania State University, USA
Pengfei Yu	Department of Computer Science, University of Illinois at Urbana Champaign, USA
Botao Yu	Nanjing University, China
Donglei Yu	Institute of Automation, Chinese Academy of Sciences, China
Dong Yu	Beijing Language and Culture University, China
Tiezheng Yu	The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China
Junjie Yu	Soochow University, China
Heng Yu	Shopee, China
Chunyuan Yuan	Institute of Information Engineering, Chinese Academy of Sciences, China
Xiang Yue	The Ohio State University, USA
Daojian Zeng	Hunan Normal University, China
Qi Zeng	University of Illinois at Urbana-Champaign, USA
Shuang	(Sophie)University of Oklahoma Zhai, USA
Yi Zhang	University of Pennsylvania, USA
Qi Zhang	Fudan University, China
Yazhou Zhang	Zhengzhou University of Light Industry, China
Zhuosheng Zhang	Shanghai Jiao Tong University, China
Weinan Zhang	Harbin Institute of Technology, China
Shuaicheng Zhang	Virginia Polytechnic Institute and State University, USA
Xiaohan Zhang	Institute of Automation, Chinese Academy of Sciences, China
Jiajun Zhang	Institute of Automation Chinese Academy of Sciences, China
Zhihao Zhang	Beihang University, China
Zhiyang Zhang	National Laboratory of Pattern Recognition, Institute of Automation, CAS, China
Dakun Zhang	SYSTRAN, France
Zixuan Zhang	University of Illinois Urbana-Champaign, USA
Peng Zhang	Tianjin University, China
Wenxuan Zhang	DAMO Academy, Alibaba Group, Singapore
Yunhao Zhang	National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China
Xingxing Zhang	Microsoft Research Asia, China

Yuanzhe Zhang	Institute of Automation, Chinese Academy of Sciences, China
Zhirui Zhang	Tencent AI Lab, China
Xin Zhao	Fudan University, China
Zhenjie Zhao	Nanjing University of Information Science and Technology, China
Yanyan Zhao	Harbin Institute of Technology, China
Mengjie Zhao	Center for Information and Language Processing, LMU Munich, Germany
Xinpei Zhao	CASIA, China
Sanqiang Zhao	University of Pittsburgh, Department of Informatics and Networked Systems, School of Computing and Information, USA
Wanjun Zhong	Sun Yat-Sen University, China
Ben Zhou	University of Pennsylvania, USA
Weikang Zhou	Fudan University, China
Peng Zhou	Kuaishou, China
Xin Zhou	Fudan University, China
Xiabing Zhou	Soochow University, China
Guangyou Zhou	School of Computer Science, Central China Normal University, China
Wangchunshu Zhou	ETH Zurich, Switzerland
Tong Zhu	Soochow University, China
Jie Zhu	Soochow University, China
Conghui Zhu	Harbin Institute of Technology, China
Yaoming Zhu	ByteDance AI lab, China
Muhua Zhu	Meituan Group, China

Organizers

Organized by

China Computer Federation, China

Hosted by

Guilin University of Electronic Technology

In Cooperation with

Lecture Notes in Computer Science

Springer

ACTA Scientiarum Naturalium Universitatis Pekinensis

Sponsoring Institutions

Diamond Sponsors

China Mobile

KuaiShou

OPPO

Baidu

Platinum Sponsors

GTCOM

HUAWEI

Douyin Group

Golden Sponsors

Microsoft

TRS

BaYou

NiuTrans

DATAOCEAN AI

Tencent AI Lab

Vivo

Contents – Part I

Oral: Dialogue Systems

Oral: Fundamentals of NLP

Oral: Information Extraction and Knowledge Graph

Oral: Machine Learning for NLP

Oral: Machine Translation and Multilinguality

Oral: Multimodality and Explainability

Oral: NLP Applications and Text Mining

Oral: Question Answering

Oral: Large Language Models

Oral: Summarization and Generation

Poster: Dialogue Systems

Contents – Part II

Poster: Fundamentals of NLP

Poster: Information Extraction and Knowledge Graph

Poster: Machine Learning for NLP

Poster: Machine Translation and Multilinguality

Poster: Multimodality and Explainability

Poster: NLP Applications and Text Mining

Poster: Question Answering

Poster: Large Language Models

Poster: Summarization and Generation

Contents – Part III

Poster: Summarization and Generation

Student Workshop: Information Retrieval and Text Mining

Student Workshop: Information Extraction and Knowledge Acquisition

Evaluation Workshop: Chinese Grammatical Error Correction

Evaluation Workshop: Multi-perspective Scientific Machine Reading Comprehension

Evaluation Workshop: Math Word Problem Solving

Evaluation Workshop: Conversational Aspect-Based Sentiment Quadruple Analysis

Evaluation Workshop: Chinese Medical Instructional Video Question Answering

Evaluation Workshop: Chinese Few-Shot and Zero-Shot Entity Linking

Evaluation Workshop: Chinese Essay Discourse Coherence Evaluation

Evaluation Workshop: Chinese Spelling Check

Evaluation Workshop: User Feedback Prediction and Response Generation

Evaluation Workshop: Learn to Watch TV: Multimodal Dialogue Understanding and Response Prediction

Oral: Dialogue Systems

A Task-Oriented Dialog Model with Task-Progressive and Policy-Aware Pre-training

Lucen Zhong[1], Hengtong Lu[1], Caixia Yuan[1], Xiaojie Wang[1](✉), Jiashen Sun[2], Ke Zeng[2], and Guanglu Wan[2]

[1] Center of Intelligence Science and Technology, Beijing University of Posts and Telecommunications, Beijing, China
{zhonglucen,luhengtong,yuancx,xjwang}@bupt.edu.cn
[2] Meituan, Beijing, China
{sunjiashen,zengke02,wanguanglu}@meituan.com

Abstract. Pre-trained conversation models (PCMs) have achieved promising progress in recent years. However, existing PCMs for Task-oriented dialog (TOD) are insufficient for capturing the sequential nature of the TOD-related tasks, as well as for learning dialog policy information. To alleviate these problems, this paper proposes a task-progressive PCM with two policy-aware pre-training tasks. The model is pre-trained through three stages where TOD-related tasks are progressively employed according to the task logic of the TOD system. A global policy consistency task is designed to capture the multi-turn dialog policy sequential relation, and an act-based contrastive learning task is designed to capture similarities among samples with the same dialog policy. Our model achieves better results on both MultiWOZ and In-Car end-to-end dialog modeling benchmarks with only 18% parameters and 25% pre-training data compared to the previous state-of-the-art PCM, GALAXY. We make our code and data publicly available (https://github.com/lucenzhong/TPLD).

Keywords: Task-oriented Dialog · Pre-training · Response generation

1 Introduction

Task-oriented dialog (TOD) system aims at helping users complete specific tasks through multi-turn interactions. Compared with open domain dialog agents, a TOD system generates more controllable replies by implementing three sub-tasks: 1) Dialog State Tracking (DST) extracts the belief state; 2) Dialog Policy Learning (POL) decides which acts should be taken based on the belief state; 3) Natural Language Generation (NLG) converts acts into natural language utterances. A large amount of work has been done for each sub-task [1–3] separately, as well as joint models for them [4,5].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 3–15, 2023.
https://doi.org/10.1007/978-3-031-44693-1_1

Pre-trained Conversation Models (PCMs) [9,11–13] are Pre-trained Language Models (PLMs) further pre-trained on dialog data. Although previous work on PCMs for TOD has made big progress, the following issues are still not well-addressed: 1) When TOD-related sub-tasks are used as pre-training tasks for PCMs, they are always employed simultaneously in a multi-task way. However, DST, POL and NLG are essentially sequential tasks in the TOD system. Managing sequential tasks in a multi-task way cannot capture the sequential nature of these tasks and it is difficult to better learn the subsequent task due to insufficient learning of the previous task. 2) Existing work only optimizes the policy for each dialog turn [13]. However, TOD is essentially a multi-turn sequential decision-making process, so it is more critical to build pre-training tasks that learn to optimize dialog policy over the whole dialog. In addition, existing work only models the policy differences between samples in the same batch, ignoring the similarities among samples with the same policy in data sets.

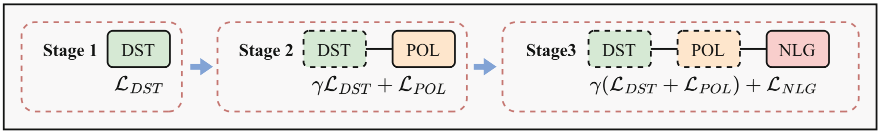

Stage 1
DST
$\mathcal{L}_{DST}$
Stage 2
DST
POL
$\gamma\mathcal{L}_{DST} + \mathcal{L}_{POL}$
Stage3
DST
POL
NLG
$\gamma(\mathcal{L}_{DST} + \mathcal{L}_{POL}) + \mathcal{L}_{NLG}$

Fig. 1. The TPLD multi-stage pre-training framework.

To address the above problems, this paper first proposes a **T**ask-**P**rogressive with **L**oss **D**ecaying (TPLD) multi-stage pre-training framework for training TOD PCMs. As shown in Fig. 1, the framework includes three stages of pre-training. DST, POL, and NLG tasks are progressively employed in different stages according to the task logic of the TOD system. Since DST, POL, and NLG tasks for PCMs are heterogeneous tasks, the latter task may completely offset the tasks in the previous stages. Therefore, tasks employed in previous stages are assigned a decayed loss in the current stage. The decayed loss is used to leverage tasks from the previous stage so that current tasks can not compeletely offset the previous task. At the same time, we propose two policy-aware pre-training tasks to enhance policy learning. A global policy consistency task, which minimizes the L_2 distance between the policy prior and policy posterior both at the turn-level and the session-level, is proposed to model both the single and multiple turn policy. We also propose an act-based contrastive learning task by introducing out-of-batch positive samples to learn the similarities between dialogs with the same policy and the differences between dialogs with different policies simultaneously.

T5-small [14] is employed as the backbone model. Experimental results show that our model outperforms previous state-of-the-art PCM on both MultiWOZ and In-Car end-to-end dialog modeling benchmarks. In summary, the main contributions of the paper are as follows:

1. We propose a task-progressive pre-training framework for TOD PCMs, which leverages sequential nature between the different pre-training tasks.
2. We propose two novel and effective policy-aware pre-training tasks for dialog policy modeling. To the best of our knowledge, it is the first session-level dialog policy pre-training task.
3. Our model achieves better results on two popular end-to-end dialog modeling benchmarks with fewer parameters and less pre-training data compared with previous strong PCMs.

2 Related Work

Pre-trained Language Models for TOD. Pre-trained Language Models (PLMs) trained on large general text corpora [14,21], have been widely applied to dialog systems [6,7]. UBAR [6] evaluates the task-oriented dialog system in a more realistic setting, where its dialog context has access to user utterances and all generated content. Mars [7] proposes two contrastive learning strategies to model the dialog context and belief/action state. Since the intrinsic linguistic patterns differ between dialog and normal text, PLMs-based TOD models achieve limited progress.

Pre-trained Conversation Models. In order to bridge the gap caused by pre-training data, some studies [22,23] further pre-trained the PLMs on dialog corpora to build pre-trained conversation models(PCMs). Many PCMs are trained on open-domain dialog data for response generation, and here we concentrate on PCMs for TOD. SC-GPT [3] first exploited pre-train PLMs for the NLG module in TOD systems. TOD-BERT [8] and SPACE-2 [12] trained a dialog understanding model that can accomplish tasks like intent recognition and state tracking. SOLOIST [9] pre-trained a task-grounded response generation model, which can generate dialog responses grounded in user goals and real-world knowledge for task completion. PPTOD [10] introduced a multi-task pre-training strategy that augments the model's ability with heterogeneous dialog corpora. GALAXY [11] proposed to learn turn-level dialog policy from limited labeled dialogs and large-scale unlabeled dialog corpora. SPACE-3 [13] combined GALAXY and SPACE-2 to propose a unified model for dialog understanding and generation. OPAL [24] leveraged external tools to generate TOD-like data to bridge the gap between pre-training and fine-tuning. Existing work did not explore pre-training methods other than multi-task learning and only learned turn-level policy.

3 Method

In this section, we first introduce the **T**ask-**P**rogressive with **L**oss **D**ecaying (TPLD) multi-stage pre-training framework for training TOD PCMs. Then we describe two policy-aware pre-training tasks. Figure 2 gives some overview information of our method.

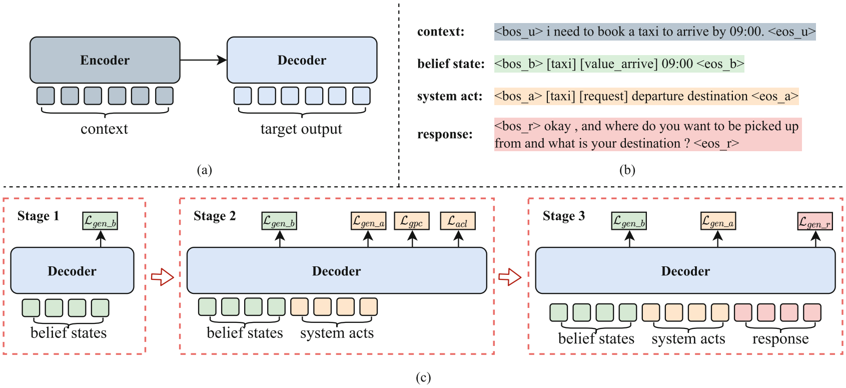


Fig. 2. Overview of our proposed method. (a) is the general T5 architecture we use as the backbone. (b) is the data format in the model. (c) is the TPLD multi-stage pre-training framework, where different color for different pre-training tasks.

3.1 TPLD Multi-stage Pre-training Framework

The pre-training process of the model is divided into three stages. DST, POL, and NLG tasks are introduced stage by stage, considering the sequential nature of these tasks in the TOD system.

Specifically, only a generative DST task that generates the belief states of dialogs is employed in the first stage. Then some POL tasks (including one generative POL task that generates the system acts) are joined in the second stage. We remain the generative DST task together with newly joined POL tasks in the second stage to prevent the model from forgetting the DST task learned in the first stage. At the same time, to make the model focus more on newly joined POL tasks, we multiply the loss function of the generative DST task by a decaying coefficient $\gamma \in [0, 1]$ to weaken its impact. Finally, the NLG task is joined in the third stage. The same decaying coefficient applies to loss functions of both the generative DST and POL tasks. NLG is naturally a generative task that generates the system response.

We give a formal description of the process as follows: We first define a general form for all three generative pre-training tasks and their loss functions. Then, we introduce the loss function stage by stage in the following subsections.

A training sample is denoted as in Eq. (1):

$$d = (c, y) \tag{1}$$

where c denotes the input dialog context, which is the concatenation of all previous utterances in the dialog. y is the target output text. It is different from different tasks. e.g., it is the belief state in the generative DST task, and the system act in the generative POL task.

Given the training sample d, the generation loss $\mathcal{L}_{gen}$ is as in Eq. (2):

$$\mathcal{L}_{gen} = \sum_{i=1}^{|y|} \log P_{\Theta} \left(y_i | y_{<i}, c \right) \tag{2}$$

where Θ is the model parameter and $y_{<i}$ indicates all tokens before i.

Stage 1: DST Pre-training. The first stage includes only one generative DST task. The output y is the belief state, and the loss is denoted as $\mathcal{L}_{gen\_b}$. The pre-training objective function for the first stage is as in Eq. (3):

$$\mathcal{L}_{stage\_1} = \mathcal{L}_{gen\_b} \tag{3}$$

Stage 2: DST+POL Pre-training. Three POL tasks are joined in the second stage. One of the POL tasks is the system act generation task, where the output y is the system act. The loss function of the task is denoted as $\mathcal{L}_{gen\_a}$. The other two POL tasks, the global policy consistency task with loss function of $\mathcal{L}_{gpc}$ and the act-based contrastive learning task with loss function of $\mathcal{L}_{acl}$, are described in Sect. 3.2 in details. The final training objective function for the second stage is as in Eq. (4):

$$\mathcal{L}_{stage\_2} = \gamma \mathcal{L}_{gen\_b} + \left(\mathcal{L}_{gen\_a} + \alpha \mathcal{L}_{gpc} + \beta \mathcal{L}_{acl} \right) \tag{4}$$

where $\gamma \in [0, 1]$ is the decaying coefficient leveraging DST and POL tasks, $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are used to leverage different POL tasks.

Stage 3: DST+POL+NLG Pre-training. The NLG task is joined in the third stage. The output y for the NLG task is the delexicalized system response, and the loss function is denoted as $\mathcal{L}_{gen\_r}$. The training objective function for the third stage is as in Eq. (5):

$$\mathcal{L}_{stage\_3} = \gamma \left(\mathcal{L}_{gen\_b} + \mathcal{L}_{gen\_a} \right) + \mathcal{L}_{gen\_r} \tag{5}$$

where γ is the same decaying coefficient as that in Eq. (4). Please note that the γ only act on the generative task.

3.2 Policy-Aware Pre-training Tasks

Global Policy Consistency Task. As shown in Fig. 3(a), we denote the output of the last token in belief states as the policy prior h^r, and the output of the last token in system acts as the policy posterior h^o. The dialog policy is unknown in the former and known in the later. Following He et al. [13], the turn-level consistency task is to minimizing the L_2 distance between the representation of the prior and the posterior:

$$\mathcal{L}_{turn} = \| h_t^r - h_t^o \|_2^2 \tag{6}$$

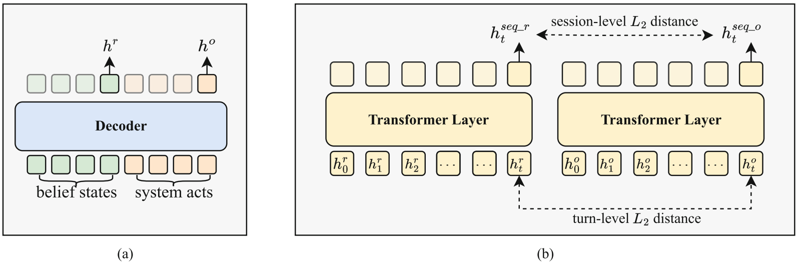

h^r
h^o
Decoder
belief states
system acts
(a)
$h_t^{seq\_r}$
session-level L_2 distance
$h_t^{seq\_o}$
Transformer Layer
Transformer Layer
h_0^r h_1^r h_2^r ... ... h_t^r
h_0^o h_1^o h_2^o ... ... h_t^o
turn-level L_2 distance
(b)

Fig. 3. (a) is the illustration of the policy prior and posterior. (b) is the illustration of global policy consistency task.

We further define the session-level loss function for training the global policy consistency task as shown in Fig. 3(b). Let the prior and the posterior of the policy vector at turn t be h_t^r and h_t^o, respectively. We can have the policy prior sequence $\{h_0^r, h_1^r, \ldots, h_t^r\}$ and the policy posterior sequence $\{h_0^o, h_1^o, \ldots, h_t^o\}$ in hand with the dialog steps forward. A single transformer layer is used to transform the policy sequence into a policy sequence representation for both prior and the posterior policy as shown in Eqs. (7) and (8):

$$h_t^{seq\_r} = Transformer\left(h_0^r, h_1^r, \ldots, h_t^r\right) \tag{7}$$

$$h_t^{seq\_o} = Transformer\left(h_0^o, h_1^o, \ldots, h_t^o\right) \tag{8}$$

The session-level consistency task is to minimizing the L_2 distance between the representation of the prior sequence and the posterior sequence:

$$\mathcal{L}_{session} = \|h_t^{seq\_r} - h_t^{seq\_o}\|_2^2 \tag{9}$$

The training objective for the global policy consistency task is the sum of turn-level and session-level objectives, as shown in Eq. (10):

$$\mathcal{L}_{gpc} = \mathcal{L}_{turn} + \mathcal{L}_{session} \tag{10}$$

The turn-level objective models the single-turn dialog policy, while the session-level objective models the multi-turn dialog policy.

Act-Based Contrastive Learning Task. The act-based contrastive learning task aims to introduce out-of-batch positive samples. We treat all samples in the same batch as negative ones and select samples with the same dialog policy from the whole dataset as positive ones. The batch size is denoted as N, given a batch of training samples $D = \{d_1, d_2, \cdots, d_N\}$, we select M positive samples for each sample d_i in this batch and get a new batch with $(M+1)N$ size. Let $I = \{1, \ldots, (M+1)\,N\}$ be the index set of the new batch. The act-based contrastive learning loss adopts the policy prior vector h^r as the sample vector h and the learning objective is defined as in Eq. (11):

$$\mathcal{L}_{acl} = -\sum_{i \in I} \sum_{j \in P_i} \log \frac{\exp\left(\sigma\left(h_i\right) \cdot \sigma\left(h_j\right)/\tau\right)}{\sum_{l \in I, l \neq i} \exp\left(\sigma\left(h_i\right) \cdot \sigma\left(h_l\right)/\tau\right)} \tag{11}$$

where P_i is a list of size M which denotes all the positive samples of sample i in the current batch. τ is a temperature hyper-parameter. The act-based contrastive learning task can learn the similarities between samples with the same dialog policy and the differences between samples with different dialog policies simultaneously.

3.3 Fine-Tuning and Inference

In the fine-tuning stage, we focus on the end-to-end dialog modeling task in the TOD system. We only use the generation task during fine-tuning, and the target output y is the concatenation of the belief state, system act, and delexicalized response. The training objective function for fine-tuning is as in Eq. (12):

$$\mathcal{L}_{fine\_tune} = \mathcal{L}_{gen\_b} + \mathcal{L}_{gen\_a} + \mathcal{L}_{gen\_r} \tag{12}$$

Note that $\mathcal{L}_{gen\_b}$ and $\mathcal{L}_{gen\_a}$ are optional since some datasets do not have corresponding semantic labels.

In the inference stage, following Yang et al. [6], we use generated system response instead of oracle system response in the context to generate the current system response.

4 Experiment Settings

4.1 Pre-training Datasets

Five existing high-quality labeled TOD datasets are used for pre-training our model, including MultiWOZ [15], KVRET [16], MSRE2E [17], Frames [18], and CamRest676 [19]. In order to reduce the label discrepancy between different datasets, we follow the unified DA taxonomy [11] to unify the dialog act annotations and use the semantic meaning of slot to unify the slot name annotations. Compared with other PCMs, our model uses the least data, with only 25% of the pre-training data compared to GALAXY.

4.2 Evaluation Tasks and Metrics

We test our model on two popular TOD benchmarks: Stanford In-Car Assistant (In-Car) dataset [16] and the MultiWOZ dataset [20]. Following previous work [6,9], the model generates delexicalized responses. `BLEU` score [25] is used to measure the response quality. For MultiWOZ, `Inform` and `Success` [15] are also reported to measure the dialog completion. A `Combined score` [26] is computed by (`Inform` + `Success`) ×0.5+ `BLEU` as an overall quality measure. In order to make a fair comparison with previous work, we adopt the standard evaluation script [27] for the evaluation of the MultiWOZ dataset. Similarly, we calculate `Match`, `SuccF1` [5], and the `Combined score` via (`Match` + `SuccF1`) ×0.5+ `BLEU` for the In-car dataset.

4.3 Baselines

We compare our model with the state-of-the-art PCMs for TOD: 1) **SOLOIST** [9] is a GPT-based model that has been further pre-trained on two TOD datasets; 2) **PPTOD** [10] is a T5-based model that has been continually pre-trained on eleven heterogeneous annotated TOD corpora; 3) **GALAXY** [11] is a UniLM-based dialog model that explicitly learns dialog policy from labeled dialogs and large-scale unlabeled dialog corpora via semi-supervised learning; 4) **SPACE-3** [13] is a unified semi-supervised pre-trained conversation model learning from large-scale dialog corpora.

4.4 Implementation Details

We employ t5-small as the backbone. In the pre-training stage, our model is trained for about 12 h on one A100 GPU. We use the Adam optimizer [29] with a learning rate of 5e−4 and a batch size of 16 for 15 epochs at each stage. For the hyper-parameters of loss coefficients, we set $\alpha = 0.1$, $\beta = 1$, and $\gamma = 0.1$, respectively. For hyper-parameters of the act-based contrastive learning task, we set $M = 2$ and $\tau = 1.0$. We removed the validation and testing set of MultiWOZ and In-car during pre-training to avoid a data breach. In the fine-tuning stage, for the MultiWOZ dataset, the learning rate is 5e−4, and the batch size is 16. For the In-Car dataset, the learning rate is 1e−3, and the batch size is 32. We fine-tune the pre-trained model on each dataset for 10 epochs and select the best model based on the validation results. Our implementation is based on the Huggingface Library [28].

5 Experiment Results

5.1 Result Comparisons

As shown in Table 1, compared with other PCMs, our model achieves new state-of-the-art combined scores on both datasets, outperforms the previous SOTA by 2.0 and 1.2 points on MultiWOZ and In-Car respectively. In particular, it is worth noticing that our model surpasses GALAXY, the current best dialog policy learning PCM with explicit policy injection, by 1.9 `Success` rate and 0.3 `SuccF1` rate for MultiWOZ and In-Car, respectively. The higher dialog success rates of our model demonstrate that our model can learn better dialog policy than other models to facilitate the completion of dialog tasks.

5.2 Ablation Study

We performed ablation experiments on the MultiWOZ dataset, the ablation results are shown in Table 2. `w/o pre_training` means directly fine-tuning T5 on the downstream task without TOD pre-training. The results show that the proposed pre-training method brings 4 points of improvements for the MultiWOZ dataset. `w/o` *TPLD* means the model is trained with the traditional multi-task

Table 1. The Performances on MultiWOZ and In-Car dataset (We do not compare with SPACE-3 on MultiWOZ because it did not report results on the standard MultiWOZ evaluation script.)

Model	MultiWOZ				In-Car			
	Inform	Success	BLEU	Comb	Match	SuccF1	BLEU	Comb
SOLOIST	82.3	72.4	13.6	90.9	–	–	–	–
PPTOD	83.1	72.7	18.2	96.1	–	–	–	106.0
GALAXY	85.4	75.7	**19.6**	100.2	85.3	83.6	23.0	107.5
SPACE-3	–	–	–	–	85.2	83.1	22.9	107.1
ours	**89.5**	**77.6**	18.7	**102.2**	**86.2**	**83.9**	**23.6**	**108.7**

Table 2. Ablation results on MultiWOZ.

Model	Inform	Success	BLEU	Comb
ours	89.5	77.6	18.7	102.2
w/o pre_training	86.6	72.3	18.5	98.0
w/o $TPLD$	86.8	73.9	18.4	98.8
w/o $\mathcal{L}_{acl}$	87.7	75.9	18.8	100.6
w/o $\mathcal{L}_{session}$	87.1	74.8	19.0	99.9
w/o $\mathcal{L}_{gpc}$	87.4	74.6	18.6	99.6
w/o $\mathcal{L}_{acl} - \mathcal{L}_{gpc}$	86.2	74.6	19.1	99.5

learning method, which learns all pre-training tasks simultaneously. The pre-training loss is defined in Eq. (13). The Combined score reduced from 102.2 to 98.8 after removing the task-progressive training framework, which indicates that the proposed TPLD multi-stage pre-training framework is crucial for dialog modeling. It is also more difficult for multi-task learning method to optimize the parameters compared to TPLD multi-stage method.

$$\begin{aligned} \mathcal{L}_{multi\text{-}task} &= \mathcal{L}_{gen} + \alpha\mathcal{L}_{gpc} + \beta\mathcal{L}_{acl} \\ \mathcal{L}_{gen} &= \mathcal{L}_{gen\_b} + \mathcal{L}_{gen\_a} + \mathcal{L}_{gen\_r} \end{aligned} \tag{13}$$

For the ablation of the policy-aware pre-training tasks, the combined score decreases by 1.6, 2.3, and 2.6 points after removing $\mathcal{L}_{acl}$, $\mathcal{L}_{session}$, and $\mathcal{L}_{gpc}$, respectively. The model performance further decreases when removing both tasks. The results demonstrate that the two proposed policy-aware pre-training tasks can help the model learn better dialog policy to complete a dialog successfully.

5.3 Loss Decaying Coefficient Analysis

Figure 4 shows the effect of different decaying coefficients γ, where γ ranges from 0 to 1. The model has the worst and the second worst performance at $\gamma = 0$ and

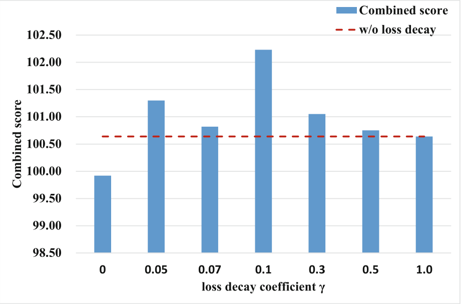
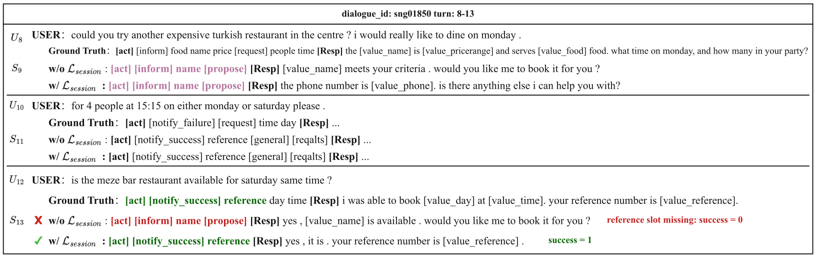


Fig. 4. The analysis of the combined score of different loss decay coefficients.



dialogue_id: sng01850 turn: 8-13

U_8 **USER**: could you try another expensive turkish restaurant in the centre ? i would really like to dine on monday .

Ground Truth: **[act]** [inform] food name price [request] people time **[Resp]** the [value_name] is [value_pricerange] and serves [value_food] food. what time on monday, and how many in your party?

S_9 **w/o** $\mathcal{L}_{session}$: **[act] [inform] name [propose] [Resp]** [value_name] meets your criteria . would you like me to book it for you ?

w/ $\mathcal{L}_{session}$ **: [act] [inform] name [propose] [Resp]** the phone number is [value_phone]. is there anything else i can help you with?

U_{10} **USER**: for 4 people at 15:15 on either monday or saturday please .

Ground Truth: **[act]** [notify_failure] [request] time day **[Resp]** ...

S_{11} **w/o** $\mathcal{L}_{session}$: **[act]** [notify_success] reference [general] [reqalts] **[Resp]** ...

w/ $\mathcal{L}_{session}$ **: [act]** [notify_success] reference [general] [reqalts] **[Resp]** ...

U_{12} **USER**: is the meze bar restaurant available for saturday same time ?

Ground Truth: **[act] [notify_success] reference** day time **[Resp]** i was able to book [value_day] at [value_time]. your reference number is [value_reference].

S_{13} ✗ **w/o** $\mathcal{L}_{session}$: **[act] [inform] name [propose] [Resp]** yes , [value_name] is available . would you like me to book it for you ? **reference slot missing: success = 0**

✓ **w/** $\mathcal{L}_{session}$ **: [act] [notify_success] reference [Resp]** yes , it is . your reference number is [value_reference] . **success = 1**

Fig. 5. The 8th turn to 13th turn from the dialog session SNG01850 in the test set.

$\gamma = 1$. The model achieves the best performance at $\gamma = 0.1$. It demonstrates that the proposed task-progressive with proper loss decaying multi-stage pre-training framework is effective for learning heterogeneous TOD tasks.

5.4 Case Study

Turn-Level vs. Session-Level. Figure 5 shows several output cases of the model with or without $\mathcal{L}_{session}$. The model with $\mathcal{L}_{session}$ avoids generating repetitive system acts to complete the user requests in shorter dialog turns. In the dialog, the user wants to reserve a restaurant. At turn S_9, the model with $\mathcal{L}_{session}$ or without $\mathcal{L}_{session}$ both `[propose]` to make a reservation for the user. At turn U_{12}, the user modifies the slot value of the restaurant name. The model with $\mathcal{L}_{session}$ knows that the reservation request has been made to the user at turn S_9, and the user agrees to make the reservation. Therefore, the reservation would be made directly at turn S_{13} and provide `reference`. However, the model without $\mathcal{L}_{session}$ does not know what system act has been generated before. Therefore, the model will repeat the same system act as turn S_9, which will miss the chance to provide `reference` to the user, and the dialog will fail.

6 Conclusion

This paper proposes a novel TPLD multi-stage pre-training framework for training TOD PCMs. The TPLD framework progressively trains the DST, POL, and NLG tasks through three successive stages. We also design two policy-aware pre-training tasks as POL tasks to model the multi-turn dialog policy sequence and policy similarity between samples during pre-training, respectively. Experiments show that our model achieves new state-of-the-art results on MultiWOZ and In-Car end-to-end dialog modeling benchmarks compared with other strong PCMs. We hope that TPLD multi-stage pre-training framework and policy-aware pre-training tasks can push forward the research in the task-oriented dialog pre-training area as well as the design for Large Language Models (LLMs) for TOD.

Acknowledgements. We are grateful to the anonymous reviewers for their insightful comments and suggestions.

References

1. Tian, X., Huang, L., Lin, Y., et al.: Amendable generation for dialogue state tracking. In: Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pp. 80–92 (2021)
2. Takanobu, R., Liang, R., Huang, M.: Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 625–638 (2020)
3. Peng, B., Zhu, C., Li, C., et al.: Few-shot natural language generation for task-oriented dialog. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 172–182 (2020)
4. Madotto, A., Wu, C.S., Fung, P.: Mem2Seq: effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1468–1478 (2018)
5. Lei, W., Jin, X., Kan, M.Y., et al.: Sequicity: simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1437–1447 (2018)
6. Yang, Y., Li, Y., Quan, X.: UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 16, pp. 14230–14238 (2021)
7. Sun, H., Bao, J., Wu, Y., et al.: Mars: semantic-aware contrastive learning for end-to-end task-oriented dialog. arXiv preprint arXiv:2210.08917 (2022)
8. Wu, C.S., Hoi, S.C., Socher, R., et al.: TOD-BERT: pre-trained natural language understanding for task-oriented dialogue. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 917–929 (2020)
9. Peng, B., Li, C., Li, J., et al.: Soloist: building task bots at scale with transfer learning and machine teaching. Trans. Assoc. Comput. Linguist. **9**, 807–824 (2021)

10. Su, Y., Shu, L., Mansimov, E., et al.: Multi-task pre-training for plug-and-play task-oriented dialogue system. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4661–4676 (2022)
11. He, W., Dai, Y., Zheng, Y., et al.: Galaxy: a generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 10, pp. 10749–10757 (2022)
12. He, W., Dai, Y., Hui, B., et al.: SPACE-2: tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 553–569 (2022)
13. He, W., Dai, Y., Yang, M., et al.: Unified dialog model pre-training for task-oriented dialog understanding and generation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 187–200 (2022)
14. Raffel, C., Shazeer, N., Roberts, A., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
15. Budzianowski, P., Wen, T.H., Tseng, B.H., et al.: MultiWOZ-a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 5016–5026 (2018)
16. Eric, M., Krishnan, L., Charette, F., et al.: Key-value retrieval networks for task-oriented dialogue. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pp. 37–49 (2017)
17. Li, X., Wang, Y., Sun, S., et al.: Microsoft dialogue challenge: building end-to-end task-completion dialogue systems. arXiv preprint arXiv:1807.11125 (2018)
18. El Asri, L., Schulz, H., Sarma, S.K., et al.: Frames: a corpus for adding memory to goal-oriented dialogue systems. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pp. 207–219 (2017)
19. Wen, T.H., Vandyke, D., Mrkšić, N., et al.: A network-based end-to-end trainable task-oriented dialogue system. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 438–449 (2017)
20. Eric, M., Goel, R., Paul, S., Sethi, et al.: Multiwoz 2.1: multi-domain dialogue state corrections and state tracking baselines. arXiv preprint arXiv:1907.01669 (2019)
21. Radford, A., Wu, J., Child, R., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
22. Henderson, M., Casanueva, I., Mrkšić, N., et al.: ConveRT: efficient and accurate conversational representations from transformers. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2161–2174 (2020)
23. Adiwardana, D., Luong, M.T., So, D.R., et al.: Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977 (2020)
24. Chen, Z., Liu, Y., Chen, L., et al.: OPAL: ontology-aware pretrained language model for end-to-end task-oriented dialogue. arXiv preprint arXiv:2209.04595 (2022)
25. Papineni, K., Roukos, S., Ward, T., et al.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

26. Mehri, S., Srinivasan, T., Eskenazi, M.: Structured fusion networks for dialog. In: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, pp. 165–177 (2019)
27. Nekvinda, T., Dušek, O.: Shades of BLEU, flavours of success: the case of MultiWOZ. In: Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pp. 34–46 (2021)
28. Wolf, T., Debut, L., Sanh, V., et al.: Huggingface's transformers: state-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
29. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

Retrieval-Augmented Knowledge-Intensive Dialogue

Zelin Wang, Ping Gong(✉), Yibo Zhang, Jihao Gu, and Xuanyuan Yang

School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
{wang_zelin,pgong,zhangyibo,gujihao}@bupt.edu.cn

Abstract. Large pre-trained language models have been shown to be powerful in open-domain dialogue. However, even the largest dialogue models suffer from knowledge hallucination, generating statements that are plausible but factually incorrect. Recent works, such as RAG and FiD, have introduced retrieval methods to alleviate this issue by bringing in external knowledge sources. Based on this research direction, we propose a plug-and-play method to enhance the generator's performance by introducing demonstration-based learning, which allows the generator to better understand the task. Furthermore, we propose a novel representation-interaction ranking model, RM-BERT, inspired by residual connections, which can more effectively represent the semantic information of context and document to improve retrieval accuracy. Experimental results indicate that the generator improvement method is applicable to multiple datasets and multiple models simultaneously. Additionally, we demonstrate that RM-BERT achieves performance close to BERT while significantly reducing computational overhead.

Keywords: Dialogue system · Retrieval-augmented generation · Document retrieval

1 Introduction

Large pre-trained language models, such as GPT3 [2], PaLM [4] and ChatGPT, have been shown to produce impressive performance on a range of tasks, especially in generating coherent, fluent and human like texts [7]. Knowledge is implicitly stored in the weights of these models, which often contain billions of parameters, enabling them to have certain knowledge on open-domain topics [25]. Unfortunately, even the largest dialogue models suffer from the hallucination of knowledge, which can be interpreted as a form of lossy compression when employing training to encode that knowledge within the weights of a neural network [13]. The phenomenon of knowledge hallucination is prevalent in knowledge-intensive dialogue, which can significantly erode the user's trust in the model-generated results.

This study was supported by the National Natural Science Foundation of China under Grant 51978300.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 16–28, 2023.
https://doi.org/10.1007/978-3-031-44693-1_2

Recent works, such as RAG [16] and FiD [9], have augmented model generation by integrating external knowledge sources, known as retrieval-augmented text generation. This approach effectively mitigates the problem of knowledge hallucination. Retrieval-augmented text generation is a new text generation paradigm that fuses emerging deep learning technology and retrieval technology [17]. Compared with generation-based counterpart, this new paradigm not only improves the correctness of the model to generate factual answers, but also helps avoid safe but boring responses by leveraging external knowledge that is not present in the dialogue history. Besides, without significantly affecting model performance, the knowledge is not necessary to be implicitly stored in model parameters, but is explicitly acquired in a plug-and-play manner, leading to great scalability [17]. For example, Retro [1] obtains comparable performance to GPT-3 and Jurassic-1 [18], despite using 25× fewer parameters.

The key to improving the quality of generation is how to extract useful information from retrieved external documents and integrate them into answers. Specifically, on the dialogue response generation task, exemplar/template retrieval as an intermediate step has been shown beneficial to informative response generation [3,29,30]. Therefore, our work proposes a demonstration-based learning method, as shown in Fig. 1. We concatenate context and document with demonstration examples retrieved from the training set and feeds them into the generator. Subsequently, only the document and context segments generated by the encoder are utilized as input for the decoder to compute multi-head attention. The goal is to provide the generator with a better understanding of the task from the demonstrations so that it can make better use of the retrieved documents. As a plug-and-play module, our approach can be applied to various existing encoder-decoder models. To our knowledge, our work is the first to use demonstration-based learning to improve generator structures in knowledge-intensive dialogue.

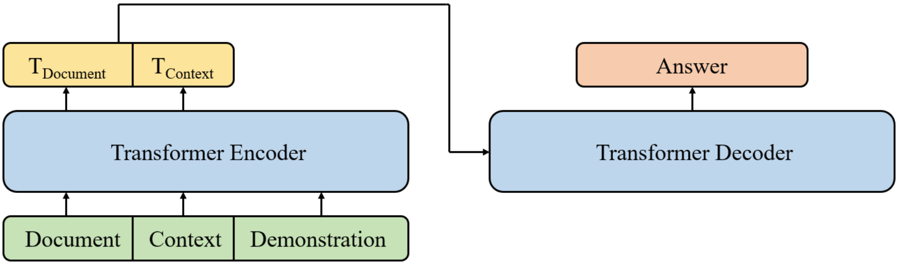


Fig. 1. Demonstration-based learning framework for generator

In knowledge-intensive dialogue, the accuracy of the retrieval model plays a crucial role in generating subsequent responses. [20] demonstrated for the first time from both theoretical and experimental perspectives that when the document is longer, it is difficult for a single fixed-dimensional vector to effectively represent the semantic information of the document. To obtain better results, it is necessary to expand the dimension of the representation vector or increase the number of representation vectors. In this paper, we propose **RM-BERT**, a representation-interaction ranking model based on **r**esidual **m**ulti-vector over **BERT**, as shown in Fig. 2. By increasing the number of representation vectors, RM-BERT can better represent the semantic information of both documents and contexts, thereby improving retrieval performance. While conceptually simple, RM-BERT outperforms strong baselines on several benchmarks and achieves similar performance to BERT, while computing much faster than BERT.

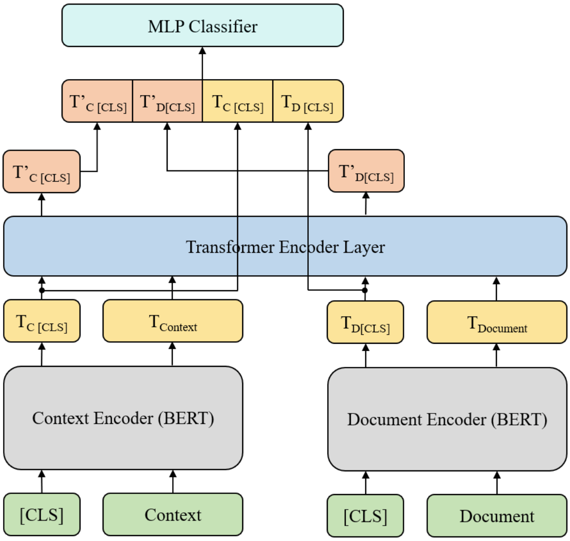


Fig. 2. Architecture of the RM-BERT

2 Related Work

Dense Retriever. According to the different ways of encoding the context and document as well as of scoring their similarity, dense retrievers can be roughly divided into three types [31]. Representation-based models, such as Realm [8] and DPR [11], are known for their speed, as the representations of documents can be computed and indexed offline in advance. However, the accuracy of these models is often compromised, as they only capture shallow interactions. Interaction-based models, such as BERT [5], can offer high accuracy by allowing for deep interactions between context and document. However, the heavy computation required by these models is not practical for real-world applications. To strike a balance between effectiveness and efficiency, a representation-interaction model is preferred. Many of these models use a late interaction architecture, such as ColBERT [12]. This architecture can not only compute the document representation in advance but also allow deep interactions between context and document, thus offering a good trade-off between speed and accuracy.

Generator. Recent research has focused on how to train a generator to better use retrieved external knowledge to improve the quality of the generated output. In previous studies, most models such as RAG and FiD just concatenate document and context and then input them into the model. They hope that the model will directly learn how to use the retrieved documents during the gradient descent. In contrast to these approaches, our proposal uses demonstration-based learning. In the supervised learning setting, the text most similar in distribution to the data in inference is the training data [28]. Therefore, we retrieve data from the training set as demonstration examples. By providing appropriate training data as demonstration examples, the model can gain a better understanding of the task and produce improved answers.

3 Methodology

Consider a conditional generation task where the input is a context c and the answer y is a sequence of tokens. To achieve better generation results in knowledge-intensive dialogue, external documents $D = \{d_1, d_2, ..., d_n\}$ are usually introduced. Retrieval-augmented text generation is the approach of adding external documents D for the model to condition on during its generation of y.

First, the retriever retrieves the top-k documents $D_k = \{d_1, d_2, ..., d_k\}$ most relevant to context c. Retriever can either directly retrieve the entire documents, or perform a preliminary retrieval first, and then sort the results. The generator then takes the context c and documents D_k as input to predict the answer y. Our generator is an encoder-decoder architecture designed according to the sequence-to-sequence modeling paradigm [26]. Specifically, the generator is based on the Transformer [27] architecture (e.g., BART [15] and T5 [23]) and parametrized by θ. We use the generator to model $p(y \mid c, D_k; \theta)$, where c and D_k are encoded by the bidirectional encoder, and the decoder predicts y autoregressively (conditioned on the encoded c and D_k and its left context). The likelihood of $p(y \mid c, D_k; \theta)$ is defined as:

$$p\left(y \mid c, D_k; \theta\right) = \prod_{t=1}^{N} p\left(y^t \mid y^{1:t-1}, c, D_k; \theta\right) \tag{1}$$

where N is the number of answer tokens.

Next, we introduce our generator improvement method and ranking model RM-BERT.

3.1 Generator

An illustration of our demonstration-based learning framework for generator is shown in Fig. 1. In the traditional retrieval-augmented generation, only document and context are input into the encoder. As an improvement we add additional demonstration examples as input to the encoder. In contrast to existing approaches that require additional human effort to generate such auxiliary supervisions, our demonstrations can be automatically constructed by retrieving appropriate data from the training set. More precisely, we use BM25 [24] to retrieve training instances from the training set. Following the encoding and semantic interaction of the above three by the encoder, only the document and context portions produced by the encoder are utilized as input for the decoder to compute multi-head attention.

For a given training set $T = \{(c_1, y_1), ..., (c_m, y_m)\}$, we index it into a list of key-value pairs, where c_i is the context and y_i is the ground-truth label. Given a context c_i, we search for the top-q most similar contexts in the index as demonstrations G_q. Note that during training, as the context c_i is already indexed, we filter it from the retrieval results to avoid data leakage. Then we concatenate demonstrations with context c_i and retrieved external documents $D_k = \{d_1, d_2, ..., d_k\}$ to feed into the encoder. In the multi-turn dialogue scene, we only use the answer as demonstration, so $G_q = \{y_{i_1}, ..., y_{i_q}\}$ and the input form is $[d_j; c_i; y_{i_1}; ...; y_{i_q}]$. For the single-turn dialogue scene similar to question answering, we concatenate the context and answer as demonstration, so $G_q = \{(c_{i_1}, y_{i_1}), ..., (c_{i_q}, y_{i_q})\}$ and its input form is $[d_j; c_i; c_{i_1}; y_{i_1}; ...; c_{i_q}; y_{i_q}]$. d_j is one of the documents of D_k, so the actual input is a batch of size k. After adding the demonstration examples, Eq. 1 is rewritten as:

$$p\left(y_i \mid c_i, D_k, G_q; \theta\right) = \prod_{t=1}^{N} p\left(y_i^t \mid y_i^{1:t-1}, c_i, D_k, G_q; \theta\right) \tag{2}$$

where N is the number of answer tokens. The final loss is defined as the negative log-likelihood:

$$L_{NLL} = -\log p\left(y_i \mid c_i, D_k, G_q; \theta\right) \tag{3}$$

During inference, we will not filter any retrieved information, as all the retrieve data only come from training set.

3.2 RM-BERT

As Fig. 2 illustrates, the lower portion of RM-BERT is a representation-based model similar to DPR. However, unlike DPR, the two encoders we employ share parameters. Then we extend the representation-based model by adding a Transformer encoder layer. The motivation for this is that [21] has shown that the lower biLM layers specialize in local syntactic relationships, allowing the higher layers to model longer range relationships such as coreference, and to specialize for the language modeling task at the top most layers. To ensure retrieval efficiency, we only use one layer of Transformer encoder. Finally, we add a MLP classifier behind the Transformer encoder layer for classification.

During training, we first use two encoders to capture the local syntactic relationships of context and document respectively. Then we concatenate the outputs of these two encoders and feed them into the subsequent Transformer encoder layer to model longer range relationships. To represent semantic information more fully, we concatenate $T_{C[CLS]}$ and $T_{D[CLS]}$ output by the context encoder and document encoder with $T^{'}_{C[CLS]}$ and $T^{'}_{D[CLS]}$ output by the

Transformer encoder layer. Finally, they are fed into the MLP classifier for classification. $T_{C[CLS]}$ and $T_{D[CLS]}$ focus on each individual sentence and play a role in expanding semantic information. $T^{'}_{C[CLS]}$ and $T^{'}_{D[CLS]}$ focus on the semantic information after interaction and play a role in fully understanding the relationship between context and document. Given a collection of contexts $C = \{c_1, c_2, ..., c_m\}$ and a collection of documents $D = \{d_1, d_2, ..., d_n\}$, the relevance score of c_i to d_j, denoted as $s(c_i, d_j)$, is estimated via a MLP classifier:

$$s\left(c_i, d_j\right) = MLP\left(Concat\left(T^{'}_{C[CLS]}, T^{'}_{D[CLS]}, T_{C[CLS]}, T_{D[CLS]}\right)\right) \tag{4}$$

The probability of a document d_j being relevant to the context c_i is calculated as:

$$p\left(d_j \mid c_i, D\right) = \frac{exp\left(s\left(c_i, d_j\right)\right)}{\sum_{k=1}^{|D|} exp\left(s\left(c_i, d_k\right)\right)} \tag{5}$$

The ranking model parameters are updated by minimizing the cross-entropy loss:

$$L = -\sum_{(c_i, d_j)} \left(z_{i,j} log\left(p_{i,j}\right) + \left(1 - z_{i,j}\right) log\left(1 - p_{i,j}\right)\right) \tag{6}$$

where $z_{i,j}$ is the ground-truth label of c_i and d_j and $p_{i,j}$ is equivalent to $p(d_j \mid c_i, D)$.

During inference time, we first apply the document encoder to all the documents and save the encoding results offline. At runtime, only the context needs to be fed into the context encoder for encoding. The context encoding result is then fed into the Transformer encoder layer together with the precomputed document encoding result.

4 Experiments

In this section, we will introduce more details about experiments and the corresponding analysis.

4.1 Datasets

We conduct experiments on three datasets: Wizard of Wikipedia [6], Natural Questions [14] and TriviaQA [10]. Wizard of Wikipedia is a large dataset of

Table 1. Dataset statistics

Task	Dataset	Train	Dev
multi-turn dialogue	WoW	63734	3054
single-turn dialogue	NQ	87372	2837
single-turn dialogue	TriviaQA	61844	5359

multi-turn dialogue grounded with knowledge retrieved from Wikipedia. The input is a short dialog history ending with the information seekers turn. Natural Questions and TriviaQA are question answering datasets, which are equivalent to single-turn dialogue scenario in knowledge-intensive dialogues. Unlike the original versions, the relevant Wikipedia page must be found by a retrieval step. Overall statistics can be found in Table 1.

To retrieve the necessary information, we employ the standard KILT Wikipedia dump[1] [22]. Without loss of generality, we only merge the ground-truth label documents of the three datasets as our knowledge source for retrieval.

4.2 Evaluation Metrics

We employ standard KILT automatic metrics. KILT contains multiple evaluation metrics, which can be roughly divided into three types: (1) downstream results, (2) performance in retrieving relevant evidence to corroborate a prediction and (3) a combination of the two [22].

For retrieval task, in order to measure the correctness of the provenance, we adopt R-Precision and Recall@5. R-precision, calculated as r/R, where R is the number of Wikipedia pages inside each provenance set and r is the number of relevant pages among the top-R retrieved pages. Recall@k, calculated as w/n, where n is the number of distinct provenance sets for a given input and w is the number of complete provenance sets among the top-k retrieved pages [22].

For generation task, Wizard of Wikipedia uses Rouge-L, F1, KILT-RL and KILT-F1 to measure the correctness of the generated output. Natural Questions and TriviaQA use EM, F1, KILT-EM, KILT-F1. The KILT scores only award EM, ROUGE-L and F1 points to KILT-EM, KILT-RL and KILT-F1 respectively, if the R-precision is 1 [22]. This metric is employed to emphasize the importance of systems being able to substantiate their output with appropriate evidence, rather than just providing an answer.

4.3 Implementation Details

The BM25 library we use is based on Anserini[2]. Our models training is based on Transformers library[3]. All models use the AdamW optimizer [19] and fine-tune on each dataset independently after loading the pre-trained weights.

Retrieval. BM25 and DPR directly retrieve from the entire knowledge source, while ColBERT, BERT and our RM-BERT act as rerankers to rerank the top 100 results retrieved by BM25. For DPR and ColBERT, we use the same setting as the original papers. For BERT, we employ the BERT base model. Both encoders for RM-BERT are also BERT base models, and they share parameters. The transformer encoder layer of RM-BERT is initialized using the top layer weights

[1] https://github.com/facebookresearch/KILT.

[2] https://github.com/castorini/anserini.

[3] https://github.com/huggingface/transformers.

of pre-trained BERT. For the MLP classifier, we use two linear layers with a ReLU activation function for nonlinear transformation.

Generation. To show the efficacy of demonstration-based learning as a plug-and-play method, we present performance in two models: RAG and FiD. Specifically, we use the RAG-Token model, and for FiD we use FiD-base. Referring to the original paper's settings, RAG uses 5 retrieved documents, and FiD uses 20 retrieved documents. For the Wizard of Wikipedia dataset, we use one demonstration, while for the Natural Questions and TriviaQA datasets, we use three demonstrations. We also consider demonstrations of other quantities and compare them empirically in Sect. 4.5. In addition, we also select BART and T5 as baselines, two models that do not retrieve knowledge sources, and their experimental results are sourced from [22].

4.4 Results

Table 2. Results for Retrieval

Model	WoW		NQ		TriviaQA	
	R-Prec	Recall@5	R-Prec	Recall@5	R-Prec	Recall@5
Bm25	37.07	61.23	49.10	65.04	51.37	69.44
DPR	40.86	65.62	61.02	69.69	60.94	68.28
ColBERT	44.47	70.37	65.84	76.56	68.43	79.42
RM-BERT	46.30	71.25	67.04	77.12	68.67	79.01
BERT	47.51	72.43	68.77	78.21	71.21	81.41

Retrieval. Table 2 illustrates the retrieval evaluation results of our proposed model in comparison with the baselines. The results indicate that our model achieves performance close to BERT across all datasets, while requiring considerably less computational resources. Furthermore, in comparison with ColBERT, which is also a representation-interaction model, our approach exhibits superior performance on nearly all datasets. Despite BM25 and DPR's advantage in terms of speed, they significantly lag behind our model concerning retrieval effectiveness. Overall, our model achieves an optimal trade-off between efficacy and efficiency.

Generation. In Tables 3, 4 and 5, we present the performance evaluation results of our proposed approach and the baselines. Firstly, we observe that the inclusion of the demonstration-based learning approach leads to substantial improvements in nearly all evaluation metrics across all datasets for both RAG and

Table 3. The results on Wizard of Wikipedia dataset

Model	Rouge-L	F1	KILT-RL	KILT-F1
BART	12.05	13.35	0	0
T5	12.80	13.28	0	0
RAG	16.03	18.10	10.61	12.00
FiD	14.81	16.73	9.79	11.07
RAG-demo	16.64(+0.61)	18.87(+0.77)	10.71(+0.10)	12.20(+0.20)
FiD-demo	15.70(+0.89)	17.56(+0.83)	9.85(+0.06)	11.13(+0.06)

Table 4. The results on Natural Questions dataset

Model	EM	F1	KILT-EM	KILT-F1
BART	26.15	32.06	0	0
T5	25.20	31.88	0	0
RAG	46.14	53.94	40.25	45.46
FiD	48.78	57.00	38.53	44.47
RAG-demo	47.83(+1.69)	55.41(+1.47)	41.63(+1.38)	46.83(+1.37)
FiD-demo	51.50(+2.72)	59.68(+2.68)	41.17(+2.64)	46.98(+2.51)

Table 5. The results on TriviaQA dataset

Model	EM	F1	KILT-EM	KILT-F1
BART	32.54	39.58	0	0
T5	25.79	33.72	0	0
RAG	49.60	61.92	40.57	48.80
FiD	51.26	65.87	39.39	49.50
RAG-demo	53.41(+3.81)	64.69(+2.77)	42.30(+1.73)	49.60(+0.80)
FiD-demo	54.75(+3.49)	64.23(-1.64)	42.02(+2.63)	48.52(−0.98)

FiD models, as shown in the corresponding improvements indicated in parentheses. This finding not only confirms the efficacy of our proposed approach but also demonstrates its applicability to a wide range of models. Additionally, the results indicate the usefulness of our method for both single-turn and multi-turn knowledge-intensive dialogues. Next, we compare the performance of models with and without the usage of knowledge sources. We observe a significant improvement in model performance after incorporating knowledge sources, as evidenced by the comparison results. This highlights the critical role of knowledge retrieval in knowledge-intensive dialogues.

4.5 Ablations

To explore the effect of RM-BERT, we conduct relative ablation studies, as illustrated in Table 6. The RM-BERT-single model feeds only a single vector $T'_{C[CLS]}$ into the MLP for classification. Our findings suggest that reducing the number of representation vectors resulted in decreased model performance across all datasets, with the most significant impact observed on the R-Prec metric for the Wizard of Wikipedia dataset. These results underscore the inadequacy of using only one vector to represent the semantic information of sentences and demonstrate the efficacy of our proposed method in addressing this limitation.

We also conduct ablation studies to investigate the impact of the number of demonstrations on the model's performance. The results of these studies are presented in Table 7. Our findings indicate that, in the case of multi-turn dialogue, the model's performance is optimal when only one demonstration is used. As the number of demonstrations increases, the model's performance experiences a slight decrease. Conversely, for single-turn dialogue, the model's performance is best when three demonstration examples are used, with any additional demonstration examples leading to a degradation in performance.

Table 6. Ablation experiment for the variation of RM-BERT

Model	WoW		NQ		TriviaQA	
	R-Prec	Recall@5	R-Prec	Recall@5	R-Prec	Recall@5
RM-BERT-single	40.47	71.09	66.13	76.81	65.07	78.32
RM-BERT	46.30	71.25	67.04	77.12	68.67	79.01

Table 7. The effect of different number of demonstrations on the performance of FiD-demo model

Demo num	WoW				NQ			
	Rouge-L	F1	KILT-RL	KILT-F1	EM	F1	KILT-EM	KILT-F1
1	15.70	17.56	9.85	11.13	49.24	57.64	39.27	45.31
2	15.48	17.15	9.71	10.84	50.90	59.08	40.54	46.41
3	15.12	16.69	9.30	10.36	51.50	59.68	41.17	46.98
4	15.14	16.61	9.27	10.31	51.22	59.50	41.03	46.82
5	15.00	16.41	9.10	10.09	51.22	59.37	40.89	46.56

4.6 Case Study

We conduct a case study on the WoW and NQ dev sets to intuitively compare our model with the baseline, as presented in Table 8. On the WoW task, FiD generates incorrect answer due to insufficient understanding of the context, while on the NQ task, it directly generates factually incorrect answer. In contrast, FiD-demo generates more specific and factually accurate responses.

Table 8. Case study on the WoW and NQ dev sets

Task	Input	Model	Output
WoW	User: My favorite color is red do you like it? Bot: I like red, but pink is my favorite it is named after a flowering plant in the genus Dianthus. User: What is the wavelength of red?	FiD	It was first used as a color name in the late 17th century.
		FiD-demo	I'm not sure, but I know it is the color at the end of the visible spectrum of light, next to orange and opposite violet.
NQ	Who has made the most premier league appearances?	FiD	Gary Speed
		FiD-demo	Gareth Barry

5 Conclusions

In this paper, we present a new method to improve the generator in knowledge-intensive dialogue and propose a ranking model RM-BERT. We propose to improve the generator through demonstration-based learning, allowing the generator to better utilize the retrieved knowledge sources by enhancing model's understanding of the task. Experiments prove that, as a plug-and-play method, it is applicable not only to multiple datasets, but also to a variety of models. Furthermore, we use multiple vectors to better represent the semantic information of context and document in a way similar to residual connections. Despite having a much lower computational cost than BERT, RM-BERT achieves a performance very close to BERT, which is a significant achievement in practical application scenarios. In future work, we intend to explore combining our demonstration-based learning method with other methods, potentially providing orthogonal improvement. We also intend to further explore how to represent sentences more effectively and efficiently.

References

1. Borgeaud, S., et al.: Improving language models by retrieving from trillions of tokens. In: International Conference on Machine Learning, pp. 2206–2240. PMLR (2022)
2. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
3. Cai, D., et al.: Skeleton-to-response: dialogue generation guided by retrieval memory. arXiv preprint arXiv:1809.05296 (2018)
4. Chowdhery, A., et al.: Palm: scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)

5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., Weston, J.: Wizard of wikipedia: knowledge-powered conversational agents. arXiv preprint arXiv:1811.01241 (2018)
7. Glass, M., Rossiello, G., Chowdhury, M.F.M., Naik, A.R., Cai, P., Gliozzo, A.: Re2g: retrieve, rerank, generate. arXiv preprint arXiv:2207.06300 (2022)
8. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International Conference on Machine Learning, pp. 3929–3938. PMLR (2020)
9. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282 (2020)
10. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551 (2017)
11. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
12. Khattab, O., Zaharia, M.: ColBERT: efficient and effective passage search via contextualized late interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–48 (2020)
13. Komeili, M., Shuster, K., Weston, J.: Internet-augmented dialogue generation. arXiv preprint arXiv:2107.07566 (2021)
14. Kwiatkowski, T., et al.: Natural questions: a benchmark for question answering research. Trans. Assoc. Comput. Linguist. **7**, 453–466 (2019)
15. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
16. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv. Neural. Inf. Process. Syst. **33**, 9459–9474 (2020)
17. Li, H., Su, Y., Cai, D., Wang, Y., Liu, L.: A survey on retrieval-augmented text generation. arXiv preprint arXiv:2202.01110 (2022)
18. Lieber, O., Sharir, O., Lenz, B., Shoham, Y.: Jurassic-1: technical details and evaluation. White Paper. AI21 Labs 1 (2021)
19. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
20. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguist. **9**, 329–345 (2021)
21. Peters, M.E., Neumann, M., Zettlemoyer, L., Yih, W.T.: Dissecting contextual word embeddings: architecture and representation. arXiv preprint arXiv:1808.08949 (2018)
22. Petroni, F., et al.: KILT: a benchmark for knowledge intensive language tasks. arXiv preprint arXiv:2009.02252 (2020)
23. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
24. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends® Inf. Retr. **3**(4), 333–389 (2009)
25. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. arXiv preprint arXiv:2104.07567 (2021)

26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
28. Wang, S., et al.: Training data is more valuable than you think: a simple and effective method by retrieving from training data. arXiv preprint arXiv:2203.08773 (2022)
29. Weston, J., Dinan, E., Miller, A.H.: Retrieve and refine: improved sequence generation models for dialogue. arXiv preprint arXiv:1808.04776 (2018)
30. Wu, Y., Wei, F., Huang, S., Wang, Y., Li, Z., Zhou, M.: Response generation by context-aware prototype editing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7281–7288 (2019)
31. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., Chua, T.S.: Retrieving and reading: a comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774 (2021)

Episode-Based Prompt Learning for Any-Shot Intent Detection

Pengfei Sun, Dingjie Song, Yawen Ouyang, Zhen Wu(✉), and Xinyu Dai

National Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, China
{spf,songdj,ouyangyw}@smail.nju.edu.cn, {wuz,daixinyu}@nju.edu.cn

Abstract. Emerging intents may have zero or a few labeled samples in realistic dialog systems. Therefore, models need to be capable of performing both zero-shot and few-shot intent detection. However, existing zero-shot intent detection models do not generalize well to few-shot settings and vice versa. To this end, we explore a novel and realistic setting, namely, any-shot intent detection. Based on this new paradigm, we propose **E**pisode-based **P**rompt **L**earning (EPL) framework. The framework first reformulates the intent detection task as a sentence-pair classification task using prompt templates and unifies the different settings. Then, it introduces two training mechanisms, which alleviate the impact of different prompt templates on performance and simulate any-shot settings in the training phase, effectively improving the model's performance. Experimental results on four datasets show that EPL outperforms strong baselines by a large margin on zero-shot and any-shot intent detection and achieves competitive results on few-shot intent detection.

Keywords: Intent detection · Any-shot learning · Prompt learning

1 Introduction

Intent detection is an essential component of task-oriented dialogue systems, which can be treated as a text classification problem to predict the intent of users' text input [3]. Although supervised deep learning methods [16,20] have achieved promising performance for intent detection, their success depends on large-scale annotated data. The models' generalization ability is limited in low-resource settings [1]. Therefore, understanding how to detect users' intent in low-resource settings effectively has become an emerging research topic.

Zero-shot learning (ZSL) and few-shot learning (FSL) have recently provided feasible solutions for low-resource intent detection. Although both paradigms have similar goals - using limited annotated data (including zero or a few annotated data) to detect intent - they still require different models designed for different settings. Specifically, zero-shot intent detection (ZSID)[1] models are

P. Sun and D. Song—Equal contribution.

[1] Zero-shot intent detection is a setup in which a model can learn to detect intents that it hasn't explicitly seen before in training [21].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 29–41, 2023.
https://doi.org/10.1007/978-3-031-44693-1_3

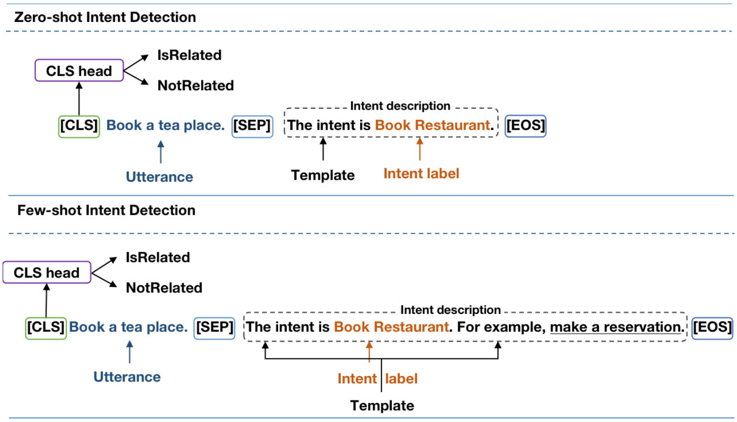

Zero-shot Intent Detection
CLS head
IsRelated
NotRelated
[CLS] Book a tea place. [SEP]
Intent description
The intent is Book Restaurant.
[EOS]
Utterance
Template
Intent label
Few-shot Intent Detection
CLS head
IsRelated
NotRelated
[CLS] Book a tea place. [SEP]
Intent description
The intent is Book Restaurant. For example, make a reservation.
[EOS]
Utterance
Intent label
Template

Fig. 1. Different settings use different templates to combine with each intent label as intent descriptions. Notably, in the few-shot intent detection, the template also includes the annotation data, which is underlined.

built on the semantic matching ability of labels. In contrast, few-shot intent detection (FSID)[2] models are built on the semantic summarization ability of a few examples [13]. This means that ZSID models are ineffective when applied to FSID and vice versa. However, in realistic scenarios, some emerging intents may have no available examples (i.e., zero-shot intents), while others may have a few (i.e., few-shot intents). For this reason, there is a need for a more realistic setting where it is possible for both zero-shot and few-shot intents can co-occur during inference. This paper refers to this setting as *any-shot intent detection* (ASID). ASID aims to develop a model that can unify ZSID and FSID in a single framework.

To achieve this goal, we intend to model around the common information (intent labels) of ZSID and FSID for unified modeling across different settings. Specifically, as illustrated in Fig. 1, we use different prompt templates for different settings to incorporate each intent label into sentences, regarded as the intent descriptions. Afterward, the intent detection task can be reformulated into a sentence-pair classification task to determine if the utterance relates to the intent description. Although this approach achieves uniform modeling across different settings, it still faces the following challenges: (1) we have empirically found that this approach relies on prompt templates' design, and (2) how the model efficiently utilizes the limited annotated data for training in the ASID task.

To address the above challenges, we propose a novel and unified **E**pisode-based **P**rompt **L**earning (EPL) framework for ASID. The framework first reconstructs the intent detection task into a sentence-pair classification task using different prompt templates to unify the different settings. After that, as shown in Fig. 2, two-stage training mechanisms are proposed to alleviate the impact

[2] Few-shot intent detection is a setup in which a model can learn to detect intents that only a few annotated examples are available [22].

of different prompt templates on performance and improve the model's generalization ability. In stage 1, we propose a prompt-based pre-training mechanism to quickly adapt the model to different prompt templates and learn transferable task-specific knowledge. Specifically, we design the supervised and self-supervised losses for continued pre-training [8] on the reconstructed dataset. In stage 2, we introduce an episodic training mechanism[3] [19] to train the model within a collection of episodes, each designed to simulate an any-shot setting. The model gradually learns to adapt to emerging intents by training in multiple episodes and improves its generalization ability. Extensive analytical experiments have shown that two-stage training mechanisms can effectively mitigate the effects of different prompt templates on performance, helping the model achieve competitive results in the any-shot setting.

The main contributions of the method presented are below:

- For the first time, we present the any-shot intent detection task where both zero-shot and few-shot intents simultaneously co-occur during inference.
- We propose a unified **E**pisode-based **P**rompt **L**earning (EPL) framework for ASID. EPL models different settings uniformly by reformulating the intent detection task as a sentence-pair classification task and proposing a two-stage training mechanism to train the model. The mechanism effectively alleviates the model's reliance on prompt templates and improves its generalization ability.
- We comprehensively evaluate our approach on four popular benchmark datasets and demonstrate the superiority of our approach in zero-shot, few-shot, and any-shot intent detection.

2 Preliminaries

2.1 Problem Statement

Suppose the intents set is $\mathcal{Y} = \mathcal{Y}_s \cup \mathcal{Y}_u, \mathcal{Y}_s \cap \mathcal{Y}_u = \varnothing$, where $\mathcal{Y}_s$ is the set of seen intents, $\mathcal{Y}_u$ is the set of unseen (emerging) intents. We take the dataset of seen intents as the training set, which is defined as $\mathcal{D}_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}_{\mathrm{tr}}|}$, where $y_i \in \mathcal{Y}_s$ and $x_i \in \mathcal{X}_s$, $\mathcal{X}_s$ is the set of seen utterances. Similarly, we take the dataset corresponding to the unseen intents as the test set, which is defined as $\mathcal{D}_{\mathrm{te}} = \{(x_j, y_j)\}_{j=1}^{|\mathcal{D}_{\mathrm{te}}|}$, where $y_j \in \mathcal{Y}_u$ and $x_j \in \mathcal{X}_u$, $\mathcal{X}_u$ is the set of unseen utterances. In ZSID, no utterances for each $y_j \in \mathcal{Y}_u$ are available during the inference phase. The goal is to train a model based on the $\mathcal{D}_{\mathrm{tr}}$ so that it can predict the label of $\mathcal{X}_u$ that belongs to unseen intents. Moreover, in FSID, only a few randomly chosen utterances for each $y_j \in \mathcal{Y}_u$ are available during the inference phase, and the goal is the same as the zero-shot setting. While in ASID, a few or no utterances for each $y_j \in \mathcal{Y}_u$ are available during the inference phase, and the goal is to predict both zero-shot and few-shot intents.

[3] Episodic training mechanism attempts to simulate a realistic setting by generating a small set of artificial tasks from a larger set of training tasks for training and proceeds similarly for testing.

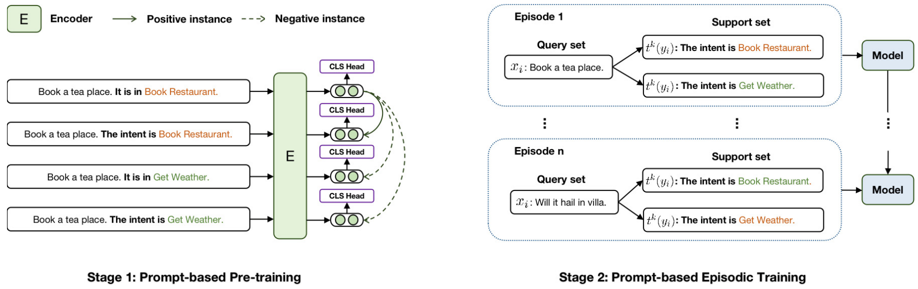

E Encoder
Positive instance
Negative instance
CLS Head
Book a tea place. It is in Book Restaurant.
Book a tea place. The intent is Book Restaurant.
Book a tea place. It is in Get Weather.
Book a tea place. The intent is Get Weather.
E
Stage 1: Prompt-based Pre-training
Episode 1
Query set
x_i: Book a tea place.
Support set
$t^k(y_i)$: The intent is Book Restaurant.
$t^k(y_i)$: The intent is Get Weather.
Model
Episode n
Query set
x_i: Will it hail in villa.
Support set
$t^k(y_i)$: The intent is Book Restaurant.
$t^k(y_i)$: The intent is Get Weather.
Model
Stage 2: Prompt-based Episodic Training

Fig. 2. The framework of the multi-stage training mechanism. Stage 1 is the prompt-based pre-training mechanism, which quickly adapts the model to different prompt templates and learns transferable task-specific knowledge. Stage 2 is the prompt-based episodic training mechanism, which enhances the model's generalization ability as the episodes progress. Note that an intent label with orange fonts indicates the instance is related (i.e., $y^{i,k}_{\text{output}} = 0$), and an intent label with green fonts indicates the instance is unrelated (i.e., $y^{i,k}_{\text{output}} = 1$). The bold text indicates the prompt template.

2.2 Episodic Training Mechanism

In the episodic training mechanism [19], an episode is similar to a batch in the traditional training method. One episode consists of two parts: support set S and query set Q. If the support set S contains C classes, and each class includes K labeled samples, such as $K = 0$, we call this task C-way K-shot. The episodic training mechanism keeps the training conditions consistent with the test conditions. For example, if the test task is a 5-way 1-shot intent detection task, our model is trained using the same 5-way 1-shot task during training. The advantage is that if a model can fit these large numbers of episodes when faced with a new, similar target task, it can also generalize to this target task.

3 Episode-Based Prompt Learning

3.1 Reformulation and Definition

Task Reformulation. We reformulate the intent detection task as a sentence-pair classification task using prompt templates. A set of prompt templates is defined as $T = \{t^k\}_{k=1}^{M}$, where M is the number of prompt templates. For each prompt template t^k, we take the intent $y_i \in \mathcal{Y}_s$ as input and output a sentence $t^k(y_i)$ which is regarded as the intent description. Based on the above definition, we use the prompt template t^k to reconstruct dataset $\mathcal{D}_{\text{tr}}$ into a sentence-pair classification dataset $\hat{\mathcal{D}}^k_{\text{tr}} = \{(x^{i,k}_{\text{input}}, y^{i,k}_{\text{output}})\}_{i=1}^{|\mathcal{D}_{\text{tr}}|}$, where $x^{i,k}_{\text{input}} = (x_i, t^k(y_i))$. The output space $y^{i,k}_{\text{output}}$ is composed of two labels, when the label of x_i is y_i, $y^{i,k}_{\text{output}} = 0$. Otherwise, $y^{i,k}_{\text{output}} = 1$. To adapt the sentence-pair classification training, we reconstructed the training dataset as $\hat{\mathcal{D}}_{\text{tr}} = \{(\hat{\mathcal{D}}^k_{\text{tr}}, -\hat{\mathcal{D}}^k_{\text{tr}})\}_{k=1}^{M}$, where $-\hat{\mathcal{D}}^k_{\text{tr}}$ denotes negative samples, i.e., the label of x_i is not y_i. The dataset $\mathcal{D}_{\text{te}}$ is similarly reconstructed to obtain $\hat{\mathcal{D}}_{\text{te}}$.

Definition. We denote the representations of $x^{i,k}_{\text{input}}$ as $\mathbf{h}^{i,k} \in \mathbb{R}^H$, which is the hidden vector of [CLS]. The output probability of the model's [CLS] head is defined as follows:

$$p_{\text{cls}}(y^{i,k}_{\text{output}}|x^{i,k}_{\text{input}}) = \text{softmax}(\mathbf{W}\mathbf{h}^{i,k}), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{2\times H}$ is a learnable matrix.

3.2 Prompt-Based Pre-training

We propose a prompt-based pre-training mechanism to quickly adapt the model to different prompt templates and learn transferable task-specific knowledge. Specifically, we use the reconstructed dataset $\hat{\mathcal{D}}_{\text{tr}}$ to pre-train the model with the supervised loss $\mathcal{L}_{\text{ce}}$ and the self-supervised loss $\mathcal{L}_{\text{con}}$.

The supervised loss $\mathcal{L}_{\text{ce}}$ is designed to narrow the gap between the pre-trained language model and the downstream tasks and to speed up the learning of transferable task-specific knowledge. We thus define the supervised loss as follows:

$$\mathcal{L}_{\text{ce}} = -y^{i,k}_{\text{output}}\log\, p_{\text{cls}}(y^{i,k}_{\text{output}}|x^{i,k}_{\text{input}}) - (1 - y^{i,k}_{\text{output}})\log(1 - p_{\text{cls}}(y^{i,k}_{\text{output}}|x^{i,k}_{\text{input}})). \tag{2}$$

Furthermore, we use different semantic information generated by different prompt templates as self-supervised signals to design the self-supervised loss function $\mathcal{L}_{\text{con}}$. It can more effectively stimulate the language model to learn to adapt to different semantic information, thus alleviating the performance difference caused by different prompt templates. Accordingly, we follow [5] and define the self-supervised loss function $\mathcal{L}_{\text{con}}$ as the normalized temperature-scaled cross-entropy loss. Concretely, for each sample $x^{i,k}_{\text{input}}$ in a batch, we treat $x^{i,k}_{\text{input}}$ as an anchor sample. We use samples $\{x^{i,n}_{\text{input}}\}^M_{n=1}(n \neq k)$ constructed from different prompt templates as positive examples, and other samples within the same batch as negative examples. The self-supervised loss function is:

$$\mathcal{L}_{\text{con}} = -\frac{1}{M-1}\log\frac{\sum_{n=1}^{M}\mathbb{I}_{[n\neq k]}\exp(\mathbf{h}^{i,k}\cdot\mathbf{h}^{i,n}/\tau)}{\sum_{j=1}^{B}\sum_{p=1}^{M}\exp(\mathbf{h}^{i,k}\cdot\mathbf{h}^{j,p}/\tau)}, \tag{3}$$

where B is the number of samples in a batch, and τ is the temperature hyperparameter. The overall loss function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda\mathcal{L}_{\text{con}}, \tag{4}$$

where λ is a hyperparameter that balances the supervised and self-supervised losses.

3.3 Prompt-Based Episodic Training

To effectively mimic low-resource settings, we introduce the prompt-based episodic training mechanism to train the ASID model. Specifically, this mechanism simulates any-shot settings by constructing a series of episodes (each episode aims to mimic an ASID task), gradually accumulating a wealth of experience in predicting unseen intents, enhancing the model's adaptability, and improving its generalization ability. In addition, we empirically find that this training mechanism further alleviates the performance differences caused by different prompt templates.

As shown in Stage 2 of Fig. 2, we first divide the training process into a series of episodes, in which each episode includes the support set S and the query set Q. We construct the support set S by designing a few prompt templates and then using them in both ZSID and FSID to eliminate the differences between the two settings. Specifically, in ZSID, the intent label y_i is mapped to the prompt template t^k as the intent description $t^k(y_i)$. Afterward, we define the intent description $t^k(y_i)$ and the corresponding output $y^{i,k}_{\text{output}}$ as a support set sample, i.e., $S_i = \{(t^k(y_i), y^{i,k}_{\text{output}})\}^{M}_{k=1}$. In FSID, we concatenate support set samples $\{x_j\}^{K}_{j=1}$ with the original prompt templates as new prompt templates $T_{\text{new}} = \{t^k\}^{M\times K}_{k=1}$. After that, same as ZSID, the intent label y_i is mapped to the new prompt template t^k as the intent description $t^k(y_i)$, so that we can get a support set that contains both the intent label and a few labeled samples, i.e., $S_i = \{(t^k(y_i), y^{i,k}_{\text{output}})\}^{M\times K}_{k=1}$. For the query set Q, the settings of $x_i \in \mathcal{X}_u$ are the same in ZSID and FSID. Since ASID sits on a continuum between ZSID and FSID, the above settings are still used for ASID. In addition, the probability of the model is defined as follows:

$$p(y_i|x_i, t^k) = \frac{\exp\, p_{\text{cls}}(y^{i,k}_{\text{output}} = 0|x_i, t^k(y_i))}{\sum_{y_j\in\mathcal{Y}} \exp\, p_{\text{cls}}(y^{i,k}_{\text{output}} = 0|x_i, t^k(y_j))}. \tag{5}$$

Finally, we define the loss $\mathcal{L}_{\text{et}}$ as follows:

$$\mathcal{L}_{\text{et}} = -\log\, p(y_i|x_i, t^k). \tag{6}$$

Notably, during inference, given an input sentence x, for each intent label $y_i \in \mathcal{Y}_s$, the model generates a set of rendered prompts $t^k(y_i)$ using different prompt templates. The model then computes a similarity score between x and each $t^k(y_i)$ using the similarity function $p_{\text{cls}}(y^{i,k}_{\text{output}} = 0|x_i, t^k(y_i))$. Finally, the model predicts the intent label with the highest similarity score as the output.

4 Experiment

4.1 Datasets

We evaluate our models on four intent detection datasets: BANKING77 [2], SNIPS [6], HWU64 [14] and CLINC150 [11]. **BANKING77** is a fine-grained intent detection dataset specific to the banking domain and is composed of 77 intents with approximately 13k samples. **SNIPS** is a personal voice assistant dataset comprising approximately 14k samples of 7 intents. **HWU64** was collected from Amazon Mechanical Turk, a fine-grained intent detection dataset.

Table 1. Statistics of four datasets (left) and templates used in our experiment (right). The *#sents* denotes sentence number, *#cls* denotes class number and *#sents/cls* denotes average sentences per class. [s] is the placeholder of the intent label.

Dataset	#sents	#cls train/valid/test	#sents/cls	Prompt Templates
SNIPS	14, 484	2/3/2	2069.14	t^1 = [**s**].
CLINC150	22, 500	50/50/50	150.00	t^2 = This is in [**s**].
BANKING77	13, 083	25/25/27	169.91	t^3 = It is about [**s**].
HWU64	11, 036	23/16/25	172.44	t^4 = It is about the intent of [**s**].

It contains approximately 11k samples of 64 intents. **CLINC150** is also a voice assistant dataset and supports both in-scope and out-of-scope data. We follow [12] for the partitioning method, using only 150 intents from 10 domains with approximately 22k samples. Zero-shot, few-shot, and any-shot settings are constructed based on these four datasets. The training, validation, and test sets are partitioned according to the intent labels without intersections. The detailed data statistics are presented in Table 1 (left).

4.2 Experimental Settings

For each experiment, we construct different episodes (i.e., C-way K-shot tasks) to evaluate the model's performance. To make a fair comparison, we complete experiments with C set to different values. For ZSID, we set C to the number of intents in the test dataset and set K to 0, i.e., 0-shot. For FSID, we set K to 1 and 5 to construct 1-shot and 5-shot scenarios, respectively. C is set to 2 on the SNIPS dataset. For the other datasets, we follow the evaluation settings from [4] and set C to 5. For ASID, we select one 0-shot class and one 1-shot class in one episode for SNIPS, one 0-shot class, two 1-shot classes and two 5-shot classes for the other datasets. Also, as the different choices of training and validation sets affect the performance of the test, we perform a 5-fold cross-validation and report the average results. In addition, in our experiments, to evaluate the impact of different templates, we manually designed four templates for the dataset and reported the average results of the four templates and the results of the best template. The detailed templates are presented in Table 1 (right).

We implement all methods using Pytorch. EPL is built on top of the BERT-base model. For the training phases of EPL, we optimize the model using Adam [10], with the stage 1 learning rate set to 5e$-$6 and the stage 2 learning rate set to 2e$-$5. For prompt-based pre-training, similar to [23], we first select the target dataset and then use the remaining three out of four datasets as the pre-training corpus. We split the pre-training corpus as the training set and validation set, where the ratio of classes is 9:1. The grid search mechanism is utilized to select optimal hyperparameter combinations on each split (the hyperparameter λ from $\{0.1, 0.5, 1, 5\}$, the negative samples' number from $\{1, 3, 5\}$). Finally, we select the hyperparameter λ and negative samples' numbers as 0.5 and 3, respectively. The temperature coefficient τ is set to 0.7, and the batch size is 8.

For prompt-based episodic training, we follow the experimental setup in [7] and report the average accuracy of over 500 episodes sampled from the test set.

4.3 Baseline Models

We compare our proposed EPL model with three types of baseline models. We first compare the EPL model with the following state-of-the-art approaches for ZSL. **CTIR**: [17] propose a class transformation framework that encourages models to learn the difference between seen and unseen intents through multi-task learning objectives and presents similarity scorers to correlate associations between intents. **NSP-BERT**: [18] propose a sentence-level prompt learning method that determines whether the two sentences are adjacent by reformulating the downstream task as a binary classification problem and directly using a BERT-based pre-trained language model.

For FSL, we compare the proposed method with several state-of-the-art FSL models. **ProtAugment**: [7] propose a meta-learning-based intent detection method that avoids the overfitting problem during meta-learning by extending the prototype network. **ContrastNet**: [4] propose a contrastive learning framework for solving the task-level and instance-level overfitting problems in the few-shot text classification task.

In addition, we also compare our approach with the models which apply in both settings. **NLI**: [15] reformulate the intent detection task as a natural language inference task and fine-tune the BERT model using the reconstructed dataset. To ensure fairness, we adapt NLI to fit ASID. **KPT**: [9] incorporate external knowledge into the verbalizers. Two strategies are designed to refine the verbalizers for the zero-shot and few-shot settings, respectively, but are not applicable for the any-shot setting.

5 Results and Analysis

5.1 Main Results

Results for ZSID. In Table 2, we report the results of the ZSID experiments. EPL outperforms other methods on all four datasets compared to state-of-the-art models. Specifically, EPL improves by 2.98%, 0.82%, 3.75%, and 0.68% on SNIPS, CLINC150, HWU64, and BANKING77, respectively. The result indicates that EPL is effective in ZSID. Notably, we also find that the EPL and NLI models are better than NSP-BERT. We speculate that retraining the [CLS] head based on the reformulated task can effectively learn transferable and task-specific knowledge, thus improving the model's performance.

Results for FSID. As shown in Table 2, we can observe that in the 1-shot setting of the FSID task, our EPL model outperforms the best baseline results (in most cases, ContrastNet). Similarly, in the 5-shot setting of the FSID task, our EPL model outperforms the best baseline results on the SNIPS, CLINC150,

Table 2. Comparison of mean accuracy (%) on four datasets in ZSID, FSID, and ASID settings. For each setting, the best and the second-best results are highlighted. The $\mathbf{Avg}_{\mathrm{Std}}$ denotes the averaged mean and standard deviation over four datasets of each model.

Setting	Method	SNIPS	CLINC150	HWU64	BANKING77	Avg_{Std}
0-shot	CTIR	86.59	27.20	21.44	20.76	$39.00_{31.86}$
	NSP-BERT	82.88	60.99	45.09	58.47	$61.86_{15.66}$
	NLI	92.20	81.51	61.01	75.45	$77.54_{13.02}$
	KPT	68.17	18.58	16.69	18.54	$30.49_{25.13}$
	EPL	**95.18**	**82.33**	**64.76**	**76.13**	$\mathbf{79.60}_{12.68}$
1-shot	ProtAugment	86.33	96.49	84.34	89.56	$89.18_{5.33}$
	ContrastNet	90.44	96.59	86.56	91.18	$91.19_{4.13}$
	NLI	94.00	95.59	83.11	90.84	$90.88_{5.55}$
	KPT	75.36	51.78	46.84	45.38	$54.84_{13.95}$
	EPL	**95.39**	**97.55**	**89.96**	**93.31**	$\mathbf{94.05}_{3.23}$
5-shot	ProtAugment	92.52	**98.74**	92.55	94.71	$94.63_{2.93}$
	ContrastNet	**96.21**	98.46	92.57	**96.40**	$\mathbf{95.91}_{2.45}$
	NLI	94.50	97.24	89.51	93.44	$93.67_{3.20}$
	KPT	80.41	62.61	50.49	47.71	$60.30_{14.88}$
	EPL	**96.21**	98.69	**92.91**	95.30	$95.78_{2.39}$
any-shot	NLI	94.00	95.50	83.34	90.53	$90.84_{5.42}$
	EPL	**95.15**	**97.47**	**88.44**	**92.78**	$\mathbf{93.46}_{3.86}$

Table 3. Impact of prompt-based pre-training on performance in ZSID. Note that the best results are highlighted.

Method		SNIPS	CLINC150	HWU64	BANKING77	Average
NSP	BERT	82.88	60.99	45.09	58.47	61.86
	Pre-training	**95.85**	75.41	59.60	70.91	75.44
EPL	BERT	94.92	81.06	59.78	74.95	77.68
	Pre-training	95.18	**82.33**	**64.76**	**76.13**	**79.60**

and HWU64 datasets and is close to the best baseline results for BANKING77. Although ContrastNet and ProtAugment are better than EPL in some datasets in the 5-shot setting, their methods only apply to FSID, which limits their application in the any-shot setting. Furthermore, we observe that KPT improves in the 1-shot and 5-shot settings compared to the zero-shot setting but still performs poorly. The results indicate that the approach resembling KPT, which relies on verbalizer construction, is less applicable to fine-grained intent detection tasks.

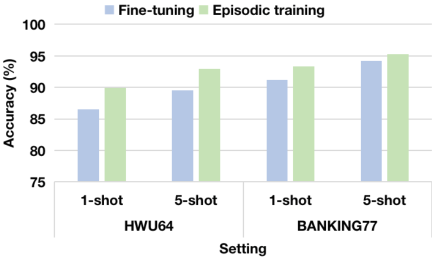
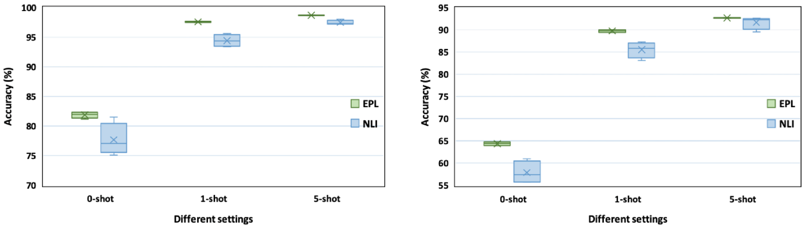


Fig. 3. Effect of different training mechanisms on two datasets.



(a) CLINC150 **(b)** HWU64

Fig. 4. Fluctuations in performance of different prompt templates on two datasets. The size of the box reflects the volatility of the data. The smaller box indicates more concentrated data. The cross represents the mean value.

Results for ASID. The results of the ASID experiments are displayed in Table 2. Compared with NLI, EPL achieves substantial improvements on the four datasets. This result validates the effectiveness of our proposed method in the any-shot setting.

5.2 Experimental Analysis

Impact of Prompt-Based Pre-training. The following experiments are conducted on four datasets to evaluate the impact of the prompt-based pre-training mechanism on performance. Concretely, we interchange the prompt-based pre-trained model and the vanilla BERT model. For NSP-BERT, we replace the vanilla BERT model with the prompt-based pre-trained model and refer to it as NSP-PreTraining. We replace the prompt-based pre-trained model in EPL with a vanilla BERT model and refer to it as EPL-BERT. Table 3 shows that using the prompt-based pre-trained model improves accuracy by an average of at least 1.9% on the zero-shot setting compared to the model using BERT. These results suggest that the prompt-based pre-training mechanism is effective. This is because continued pre-training on the reconstructed data enables the [CLS] head to adapt to downstream tasks, learn transferable task-specific knowledge, and improve the model's generalization. In addition, we note that although NSP-PreTraining achieves more competitive results on the SNIPS dataset, its accuracy is much lower than EPL on other datasets. Meanwhile, in Table 3, we can also see that EPL-BERT outperforms NSP-BERT on all datasets, which indicates that the EPL model still obtains good results even without prompt-based pre-training. This result further demonstrates that EPL is effective and superior.

Impact of Prompt-based Episodic Training Mechanism. To evaluate the impact of the prompt-based episodic training mechanism, we compare fine-tuning with the prompt-based episodic training mechanism on the HWU64 and BANKING77 datasets with different settings in FSID. As shown in Fig. 3, the model's performance based on the prompt-based episodic training mechanism outperforms that of the fine-tuning training mechanism on both the HWU64 and BANKING77, especially in the 1-shot setting, where the improvement of prompt-based episodic training is more prominent. We conjecture the prompt-based episodic training mechanism can simulate low-resource settings, and the model gradually learns to adapt to unseen intents, leading to better results. It also indicates that the prompt-based episodic training mechanism is effective.

Impact of Prompt Template. We conduct experiments to verify the effect of different prompt templates (Table 1 right) on the performance of the EPL and NLI. The results are shown in Fig. 4. Compared with NLI, EPL effectively alleviates the performance difference caused by different prompt templates, with the performance of EPL remaining stable between different prompt templates, particularly for the FSID settings. One reason could be that a few samples in the support set can be used to effectively fine-tune the model and mitigate the effects of semantic differences between different prompt templates. For the ZSID settings, the performance of both EPL and NLI fluctuates between different prompt templates, but EPL is more stable. This implies that the lack of available samples makes the model more dependent on the prompt templates, and the performance fluctuates slightly. The relative stability of EPL further suggests that EPL can mitigate the effect of differences between prompt templates.

6 Conclusion

We introduce any-shot learning to intent detection for the first time and propose a novel, unified EPL model. The EPL model first reformulates the intent detection task as a sentence-pair classification task and unifies the different settings using prompt templates. Then we propose a two-stage training mechanism (prompt-based pre-training and prompt-based episodic training). The mechanisms not only effectively mitigate the impact of different templates on performance but also simulates a low-resource setting and improve the model's generalization capability. Finally, extensive experiments have shown that the proposed model achieves state-of-the-art performance on four publicly available datasets.

Acknowledgements. The authors would like to thank the anonymous reviewers for their helpful comments. This research is supported by the National Natural Science Foundation of China (No. 61936012, 62206126 and 61976114).

References

1. Bhathiya, H.S., Thayasivam, U.: Meta learning for few-shot joint intent detection and slot-filling. In: ICMLT, pp. 86–92 (2020)
2. Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., Vulić, I.: Efficient intent detection with dual sentence encoders. In: NLP4ConvAI, pp. 38–45 (2020)
3. Celikyilmaz, A., Hakkani-Tur, D., Tur, G., Fidler, A., Hillard, D.: Exploiting distance based similarity in topic models for user intent detection. In: ASRU, pp. 425–430 (2011)
4. Chen, J., Zhang, R., Mao, Y., Xue, J.: ContrastNet: a contrastive learning framework for few-shot text classification. In: AAAI, pp. 10492–10500 (2022)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020)
6. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190 (2018)
7. Dopierre, T., Gravier, C., Logerais, W.: ProtAugment: intent detection meta-learning through unsupervised diverse paraphrasing. In: ACL/IJCNLP (2021)
8. Gururangan, S., et al.: Don't stop pretraining: adapt language models to domains and tasks. In: ACL, pp. 8342–8360 (2020)
9. Hu, S., Ding, N., Wang, H., Liu, Z., Li, J.Z., Sun, M.: Knowledgeable prompt-tuning: incorporating knowledge into prompt verbalizer for text classification. ArXiv abs/2108.02035 (2021)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
11. Larson, S., et al.: An evaluation dataset for intent classification and out-of-scope prediction. In: EMNLP-IJCNLP, pp. 1311–1316 (2019)
12. Li, J.Y., Zhang, J.: Semi-supervised meta-learning for cross-domain few-shot intent classification. In: MetaNLP (2021)
13. Liu, F., Lin, H., Han, X., Cao, B., Sun, L.: Pre-training to match for unified low-shot relation extraction. arXiv preprint arXiv:2203.12274 (2022)
14. Liu, X., Eshghi, A., Swietojanski, P., Rieser, V.: Benchmarking natural language understanding services for building conversational agents. In: IWSDS (2019)
15. Malik, V., Kumar, A., Vepa, J.: Exploring the limits of natural language inference based setup for few-shot intent detection. ArXiv abs/2112.07434 (2021)
16. Qin, L., Liu, T., Che, W., Kang, B., Zhao, S., Liu, T.: A co-interactive transformer for joint slot filling and intent detection. In: ICASSP, pp. 8193–8197 (2021)
17. Si, Q., Liu, Y., Fu, P., Lin, Z., Li, J., Wang, W.: Learning class-transductive intent representations for zero-shot intent detection. In: IJCAI (2021)
18. Sun, Y., Zheng, Y., Hao, C., Qiu, H.: NSP-BERT: a prompt-based zero-shot learner through an original pre-training task-next sentence prediction. ArXiv abs/2109.03564 (2021)
19. Vinyals, O., Blundell, C., Lillicrap, T.P., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: NIPS (2016)

20. Wang, J., Wei, K., Radfar, M., Zhang, W., Chung, C.: Encoding syntactic knowledge in transformer encoder for intent detection and slot filling. In: AAAI, vol. 35, pp. 13943–13951 (2021)
21. Xia, C., Zhang, C., Yan, X., Chang, Y., Philip, S.Y.: Zero-shot user intent detection via capsule neural networks. In: EMNLP, pp. 3090–3099 (2018)
22. Xu, W., Zhou, P., You, C., Zou, Y.: Semantic transportation prototypical network for few-shot intent detection. In: Interspeech, pp. 251–255 (2021)
23. Zhang, H., et al.: Effectiveness of pre-training for few-shot intent classification. In: EMNLP, pp. 1114–1120 (2021)

CrossDial: An Entertaining Dialogue Dataset of Chinese Crosstalk

Baizhou Huang[1,2], Shikang Du[3], and Xiaojun Wan[1,2](✉)

[1] Wangxuan Institute of Computer Technology, Peking University, Beijing, China
{hbz19,wanxiaojun}@pku.edu.cn
[2] The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing, China
[3] Ecole Polytechnique, Palaiseau, France
shikang.du@polytechnique.edu

Abstract. Crosstalk is a traditional Chinese theatrical performance art. It is commonly performed by two performers in the form of a dialogue. With the typical features of dialogues, crosstalks are also designed to be hilarious for the purpose of amusing the audience. In this study, we introduce **CrossDial**, an open-source dataset containing most classic Chinese crosstalks crawled from the Web. Moreover, we define two new tasks, provide two benchmarks, and investigate the ability of current dialogue generation models in the field of crosstalk generation. The experiment results and case studies demonstrate that crosstalk generation is challenging for straightforward methods and remains an interesting topic for future works.

Keywords: Dialogue generation · Humor · Deep learning

1 Introduction

Crosstalk, also known by its Chinese name *xiangsheng* (相声 in Chinese), is a traditional Chinese theatrical performance art. It is commonly performed by two performers. One performer is the leading role *dougen* (逗哏 in Chinese) and the other is the supporting role *penggen* (捧哏 in Chinese).

The form of crosstalks is just like chat or gossip with two persons responding to each other alternately. But there are several conventional performance patterns in crosstalks that are different from daily dialogues. First, the crosstalk is a complete story with one main topic to entertain the audience. The two performers should discuss strictly around the main topic instead of changing topics casually like gossip. Second, the language patterns of the two performers are different in crosstalks. The leading role is the one who dominates the dialogue and drives the plot forward. Mostly, the leading role tells stories and jokes during the performance. On the other hand, the supporting role gives short comments to support or question the leading role's opinion. In some cases, the supporting role may point out the humorous point in the leading role's utterance to explain to the audience, or even add fuel to the fire to make it funnier. Third, the crosstalk language is rich in comedy acting skills, such as puns, and is usually delivered in a rapid,

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 42–53, 2023.
https://doi.org/10.1007/978-3-031-44693-1_4

bantering style. For the purpose of bringing laughter to the audience, the language of crosstalk features humorous dialogues.

In the study, we are concerned with generating Chinese crosstalks automatically. Currently, the traditional art is suffering from the lack of scripts which is hard to write even for humans. It is of high artistic value to design a model that can automatically generate crosstalks. Apart from this, the ability to generate entertaining dialogue responses is also very useful in many commercial products (e.g. chat-bots) by making them more appealing. Though daily dialogue generation has been widely explored and achieved great success in previous studies [9,23], it remains unknown whether entertaining dialogues can be automatically generated or not. The special language style and the two-role pattern of crosstalks make it a challenging but interesting task to be explored.

To support research on automatic crosstalk generation, we build **CrossDial**, an open-source crosstalk dataset that covers most classic Chinese crosstalks. It is a large-scale dialogue dataset consisting of 1257 crosstalk scripts and 140432 data samples crawled from the Internet.

To investigate the automatic generation of such entertaining dialogues over the proposed dataset, we design two different tasks. The first is a generation task, i.e. **Crosstalk Response Generation**. That is, given several continuous utterances as context, the model is required to generate the next utterance as response. Considering the one-to-many problems in current generation metrics and the difficulty in automatic humor evaluation, we then additionally introduce a more basic retrieval task, i.e. **Crosstalk Response Selection**. That is, given several continuous utterances as context, the model is required to find the best response from the supported choices.

We implemented several typical neural models as baselines and evaluate them on the newly defined tasks. The results of automated metrics show the difficulty of our proposed tasks. The human evaluation and case studies further demonstrate the challenges of generating crosstalks.

The contributions of this paper are summarized as follows:

1) We propose an open-source Chinese crosstalk dataset which contains most classic Chinese crosstalks. The dataset will be released.
2) We design two different tasks and provide two benchmarks respectively for mainstream methods.
3) Both automatic evaluation and human evaluation are performed to evaluate the ability of typical models for automatic crosstalk generation.

2 Task Definition

We formulate the problem of crosstalk generation as the next utterance prediction task. In particular, we define two sub-tasks namely Crosstalk Response Generation (CRG) and Crosstalk Response Selection (CRS). Given continuous utterances as context $c = \{u_1, u_2, ..., u_{n-1}\}$, the agent is required to generate the next utterance as response $r = u_n$ in the CRG task or distinguish the positive response $x^{pos} = u_n$ from the other three distractors $\{x_0^{neg}, x_1^{neg}, x_2^{neg}\}$ in the CRS task.

In the CRG task, the generated response is expected to be grammatical and coherent to the context. It should also be compatible with the specific pattern of the role that the agent plays. Moreover, the level of amusement and humor should be taken into consideration.

CRG is a one-to-many problem, i.e. there are many responses appropriate for one given context. Most of the current generation metrics (e.g. BLEU) are based on the comparison between the reference and the generated response. Therefore they cannot reflect the true level of the agent's generation ability. As a complement, we introduce the CRS task to evaluate the agent's capability more objectively.

3 CrossDial Dataset

3.1 Overview

The objective of this work is to introduce the task of crosstalk generation and facilitate the study of both the CRG and CRS tasks. For this, we propose a large-scale Chinese crosstalk dataset **CrossDial**, which covers most classic Chinese crosstalks in the Internet. The dataset contains two subsets, **dougen** and **penggen**, corresponding to the leading role and the supporting role. Each sample for the CRG task consists of two fields: *context* and *positive response*, and each sample for the CRS task involves with three additional *negative responses (distractors)* constructed by us. The dataset creation consists of three stages:

1) Data Collection: we crawled a set of 1257 crosstalk scripts from the Internet which contains most Chinese crosstalks.
2) Sample Creation: we split all crosstalk scripts into context-response pairs as data samples for the CRG task. To be compatible with the two-role patterns in crosstalk, we divided the dataset into two subsets.
3) Distractor Generation: we designed delicate distractors for the CRS task. To avoid false negatives of distractors, we recruited eight annotators to review all the distractors and filtered invalid ones.

After all, we created a dataset consisting of 140432 data samples in the form of context-response pair. Basic statistics of **Crossdial** are shown in Table 1. In the following, we will describe all stages in more detail.

Table 1. Basic statistics of **CrossDial**. The lengths of context and response are measured in characters.

	Sample Num (trian/valid/test)	Context Len (avg/max/min)	Response Len (avg/max/min)
dougen	75944/8628/4664	302.80/3460/11	22.01/127/2
penggen	43372/5136/2688	319.20/3389/20	12.98/126/2

3.2 Data Collection

We crawled a total of $1,551$ excerpts of classic crosstalks scripts from the Internet[1]. Due to reproductions among websites, one crosstalk script might be collected from different sources. Therefore, we only kept one script and dropped the other copies. In detail, two scripts were considered the same if they have an overlap of more than 15 seven-word-longer utterances[2]. We also noticed several similar crosstalks because of a large number of script adaptations. We kept them as the status quo since they were indeed different scripts. We also took several heuristic methods for data cleaning. Finally, a total of 1257 crosstalks were collected after this process.

3.3 Sample Creation

We extracted continuous utterances from collected crosstalk scripts as context-response pairs. Specifically, for each utterance in crosstalk scripts noted as *positive response*, we extracted the sequence of no more than twenty utterances prior to it as *context*.

With the above extraction process, the response utterance in one sample may appear in the contexts of others. To avoid information leakage from the test set to the training set, we split train, validation, and test sets at the granularity of scripts instead of context-response pairs. To be exact, we randomly sampled 75 scripts for the test set, 175 scripts for the validation set, and 1007 scripts for the training set.

Considering the different speech patterns between the leading role and the supporting role, it is interesting to divide the dataset into two subsets: **dougen** and **penggen**. Each subset included only the context-response pairs where the response belonged to the corresponding role.

For the fast-paced performance before an audience, many utterances in crosstalks are designed to be short and meaningless, especially for the supporting role's lines. This is called *generic response* [8] in NLG, which may impede the diversity of dialogue systems. So we created a set of common and meaningless words. We removed data samples of which over half words of the response were in the set. We also limited the text lengths of responses to [2, 128) to avoid too long responses.

3.4 Distractor Generation

We generated distractors for the CRS task. It is both time-consuming and expensive to crowd-source human-written distractors for such a large dataset. Mostly, distractors are sampled randomly in previous work [13,24,26]. We argue that randomly sampled distractors are so simple that models may leverage shortcuts to achieve better performance. For example, random distractors commonly have fewer n-gram overlaps with context than golden response. Instead, we aim to generate high-quality distractors that are 1) similar to the golden response in order to avoid the model from using shortcuts, and 2) consistent with the semantics of context to confound the model.

[1] http://www.xiangsheng.org, http://www.tquyi.com, et al.

[2] The thresholds were set based on manual inspection of the data.

We proposed two similarity-based methods to retrieve distractors from the dataset that satisfy the above two requirements. We used the cosine distance of sentence embeddings to measure the similarity between two utterances. Pretrained language models have achieved state-of-the-art performance in the field of sentence embeddings [4–6, 17]. Given this fact, we borrowed the off-the-shelf tool, sentence transformers[3] to generate sentence embeddings for all utterances.

Response-Similar Distractor. For every data sample, we searched the corpus for similar responses as distractors with the golden response as the query. If the similarity score of the two responses is high, we consider the extracted one as a high-quality distractor.

Context-Consistent Distractor. For every data sample, we searched the corpus for similar contexts with the context of the current sample as the query and took the corresponding response as the distractor. We regard the chosen distractor to be consistent with the current context since it is the golden response to the searched context, and the two contexts resemble each other in semantics. An example of the generated distractors is shown in Table 2.

Table 2. An example of generated distractors. RSD is the abbreviation for response similar distractor and CCD is the abbreviation for context consistent distractor.

Context	您还喜欢看小说? So you enjoy reading novels?
Response	不是看而是研究, 尤其是对我国古典小说像《列国》，《水浒》，《红楼》，《西游》我都爱看，特别是对《三国演义》我敢说有独特的见解。 It's not just reading, but studying, especially our Chinese classic novels like "Chronicles of the Eastern Zhou Kingdoms", "Outlaws of the Marsh", "The Story of the Stone", and "Journey to the West". I love them all, especially "Romance of the Three Kingdoms", for which I dare say I have unique insights.
RSD	你要说三国，水浒古典名著那您研究研究，这个有点意思。 If you're talking about studying the classics like "Romance of the Three Kingdoms" and "Outlaws of the Marsh", then that's quite interesting indeed.
CCD	噢，我喜欢看的那都是古典文学呀。 Oh, what I enjoy reading is all classical literature

Extremely confusing and disorienting though the generated distractors are, it is worth considering that they can be appropriate responses. As seen in Table 2, the CCD can also be used as response to the context. To avoid the false negatives of distractors, we set a threshold λ[4] to filter all retrieved outputs that had too high cosine similarity.

[3] https://github.com/UKPLab/sentence-transformers.

[4] λ was tuning according to manual inspection of samples. Finally, we set $\lambda = 0.8$.

Apart from automatic filtering rules, we randomly drew two percent of generated distractors for quality evaluation. We found that only 5.28% of distractors were false negative. It proved that it was feasible to use generated distractors as negative responses.

In particular, for the test set, we recruited eight expert annotators to review all distractors. To simulate the process in which a model generates response conditioning on the context, we provided annotators with six rounds of prior utterances as context. Annotators were asked to select all appropriate responses from a supported choice set that is composed of generated distractors. We also added the golden response to the choice set to ensure the quality of annotation results. After annotation, we dropped invalid distractors, and randomly sampled three negative responses out of the rest for each sample.

4 Experiment

We implemented several typical models, experimented on our proposed CrossDial dataset, and provided performance benchmarks for both CRG and CRS tasks. In the following, we first introduce the baseline methods in the experiment. Then we report the results of these baselines for both CRG and CRS tasks. At last, we show the human evaluation and case studies of the generated responses.

4.1 Baseline Methods

We leveraged the open-source community huggingface[5] to build two classes of models: generative models and retrieval models. For the CRG task, we only experimented with generative models. Whereas for the CRS task, both classes were evaluated. Retrieval models take the concatenation of the context and one candidate response as input, and score it. Generative models take the context as input, and score each candidate with its generation probability. Both classes select the highest-scored candidate as the response.

In particular, we consider several generative baselines. Transformer (**Trans**) [21] is a SEQ2SEQ model which has been widely used in natural language processing. **BART** [7] is a Transformer pretrained as a denoising autoencoder[6]. **GPT** [15] is a language model pretrained with language modeling objective[7]. **T5** [16] introduced a unified framework that converts every task into a text-to-text format[8]. We also consider some retrieval baselines. **Trans**$_{enc}$ represents the encoder of Transformer model. **BERT**: BERT [2] is a Transformer encoder pretrained with masked language modeling objective[9]. **RoBERTa** [12] is built upon BERT with modified key hyper-parameters[10]. **ERNIE** [19] incorporates knowledge masking strategies to learn better language representation[11].

[5] https://huggingface.co/.

[6] BART-base, https://huggingface.co/uer/bart-base-chinese-cluecorpussmall.

[7] GPT-2, https://huggingface.co/uer/gpt2-chinese-cluecorpussmall.

[8] T5-base, https://huggingface.co/uer/t5-base-chinese-cluecorpussmall.

[9] BERT-wwm-base, https://huggingface.co/hfl/chinese-bert-wwm-ext.

[10] RoBERTa-wwm-base, https://huggingface.co/hfl/chinese-roberta-wwm-ext.

[11] ERNIE-1.0, https://huggingface.co/nghuyong/ernie-1.0.

We performed six hyper-parameters search trials for each model. Hyper-parameters and final checkpoints of baselines were both tuned with perplexity (for generative models) or accuracy (for retrieval models) on the validation set.

4.2 Result and Analysis

We trained and tested the baselines on the **penggen** and **dougen** subsets separately. The main results are reported in Table 3 and Table 4.

Table 3. Comparison of generative models on the CRG task. We adopt perplexity, BLEU [14], ROUGE [11], Distinct [8] as automated metrics.

Method	perplexity	BLEU-4	ROUGE-2	ROUGE-L	Distinct-1	Distinct-2
penggen						
Trans	14.63	3.64	1.59	15.17	1.37	6.14
BART	7.76	5.55	6.41	19.61	7.37	37.95
GPT	8.09	3.49	4.18	19.80	4.74	22.79
T5	8.80	5.75	6.71	21.75	5.47	32.54
dougen						
Trans	15.30	2.21	2.06	15.43	1.37	9.39
BART	9.41	2.98	4.19	18.00	3.69	29.16
GPT	8.75	2.24	2.35	16.25	3.00	22.30
T5	9.71	3.32	5.10	19.79	2.73	24.24

Table 4. Accuracy of various models on the CRS task. **SIM** and **CLS** are trivial methods for analysis.

	Trans	BART	GPT	T5	Trans$_{enc}$	BERT	RoBERTa	ERNIE	SIM	CLS
penggen	14.47	38.61	42.11	36.53	39.88	74.21	76.19	71.01	30.02	35.45
dougen	16.01	33.16	38.59	35.29	50.75	81.47	79.60	84.54	29.69	40.71

Penggen vs. Dougen. Dougen is the one who drives the dialogue forward, while penggen often acts as a go-between with short sentences. Intuitively, the language patterns of dougen are more complex to learn. The results on the CRG task show that the same model performs worse in the **dougen** subset than in the **penggen** subset, which is consistent with this intuition. However, experiments on the CRS task show the opposite result that retrieval models perform worse on the **penggen** subset. We attribute the phenomenon to the different difficulties of distractors of the two subsets. Since the supporting role has a relatively fixed form of response, it is more likely to retrieve high-quality distractors for the **penggen** subset, which makes it harder than the **dougen** subset.

Generative vs. Retrieval. Both generative models and retrieval models are able to handle the CRS task. The results indicate that retrieval models perform much better than generative models. It is obvious since the training objective of retrieval models is consistent with the CRS task. But still some generative models perform quite poor, especially **Trans**. The reason may be that the training of generative models is based on token-level loss while the CRS task requires a good measure of sentence-level probability to select the true response.

Shortcut Analysis. Models might use unseen patterns as shortcuts in the CRS task. We proceeded with two trivial methods, **SIM** and **CLS**, to empirically negate the phenomenon in the proposed dataset. **SIM** is an untrained method that utilizes RoBERTa to acquire sentence embeddings for the context and all candidates, and scores the candidates with the cosine similarity. **CLS** is a trained selector that takes only the candidates without context as input. We adopted the Transformer-base encoder as its backbone model. The poor performance of both methods showed that the phenomenon is not notable in the proposed dataset.

4.3 Human Evaluation

Table 5. Human evaluation of model outputs. They are rated on a scale from 0 to 3. **Gold** stands for the ground truth response.

	penggen			dougen		
	Entertainment	Readability	Relevance	Entertainment	Readability	Relevance
Gold	1.88	2.86	2.96	2.08	2.96	3.00
BART	0.76	2.94	2.10	0.82	2.94	2.22
GPT	0.48	2.80	1.54	0.64	3.00	1.86
T5	0.90	2.92	1.98	0.76	2.88	1.98
Trans	0.38	2.94	1.16	0.44	2.64	0.98

We employed two human annotators to assess 100 samples for the CRG task. Each generated response is assessed from three aspects. **Readability** measures the fluency of generated responses, including the grammar and phrase correctness; **Relevance** reflects the semantic relevance between context and response. It measures the logical and sentimental consistency of the dialogue as well; **Entertainment** reflects the level of humor of the response. For a better evaluation, we recruited two expert annotators from Tianjin, the origin of crosstalks, and familiar with the performing art of crosstalks.

Each annotator was presented with one context and five responses (including 1 golden response and 4 responses generated by different models), and asked to assign an integer score to each generated response with respect to each aspect. The scores are rated on a scale from 0(not at all) to 3(perfect without flaws).

Results are shown in Table 5. Most models can gain comparable performance with golden responses in readability. However, it can clearly be seen that generated responses

Table 6. Sampled responses generated by baseline models. The first case shows one generic response which frequently appears in generated responses. The second case shows that the model responses with a question to express surprise, which is a typical skill in crosstalk performance.

Context	逗哏 (dougen):	孙悟空那肉，它塞牙。(The meat of Monkey King will stuffed teeth.)	逗哏 (dougen):	我们亲哥儿俩跟我师父练的功夫，师父都给我们起了名字。 (My brother and I practiced kung fu with our master, and he gave us both names.)
	捧哏 (penggen):	太精瘦了。 (Too lean.)	捧哏 (penggen):	都叫什么呢？ (What are the names?)
	逗哏 (dougen):	猪八戒，太糙。 (Zhu Bajie's meat is too tough.)	逗哏 (dougen):	我哥叫“白糖的”，我叫“澄沙馅儿的”。 (My brother is called “white sugar” and I'm “red bean paste”.)
Response	捧哏 (penggen):	可不。 (Absolutely.)	捧哏 (penggen):	澄沙的？ (Red bean paste?)

are quite poor in entertainment. It indicates a huge difference between the typical models' outputs and the real crosstalks since entertainment is the most important feature of crosstalks. We can also observe that pretrained models outperform models without pretraining in all aspects by a large margin. It again proves the usefulness of pretraining.

4.4 Case Study

We make case studies to better understand the performance of models for the CRG task. Some sampled cases are shown in Table 6. We find that generated responses lack diversity, especially for models trained on **penggen**. Many generic replies appears frequently, such as “这都不像话！ (Nonsense!)” or “可不！ (Sure enough!)”. At the same time, models are unable to start a new topic, no matter whether they are trained on **dougen** or **penggen**. However, we also notice that models have learned some simple language patterns in crosstalk performance such as rhetorical patterns.

Following the initial goal to automatically generate crosstalks, we paired the best model[12] trained on **penggen** with the best one trained on **dougen**. Given the beginning of a human-written script, the two models were asked to play their corresponding roles, and generate responses alternately. All of the fifty generated crosstalks got stuck in repetitions of similar utterances after three rounds. The reason could be the lack of diversity and the similar patterns of generating responses. As a pipeline, this process also suffers from the accumulation of errors in every step.

5 Related Work

5.1 Dialogue Generation

As we mentioned above, Chinese crosstalk is a special form of dialogue. Recently, deep learning techniques have been applied to dialogue generation [9,10,18]. Among vari-

[12] According to the automated metrics, we used the T5 model in the experiment.

ous neural models, transformer [21] has proven to be one of the most effective backbone models. Building upon this, several pretrained language models utilizing the pretrain-finetune paradigm [2,7,15] have been proposed and demonstrated great success. Therefore, we chose transformer-based pretrained language models as our baselines.

The rapid expansion of social networks has facilitated the collection of large dialogue corpora from multiple data sources [1,13,20,24,26]. For example, **Cornell Movie-Dialogue Corpus** [1] collected dialogues from scripts of movies and TV series. The **Ubuntu Dialogue Corpus** [13] collected unstructured dialogues from large-scale comments on social media. These resource-based studies have significantly contributed to the progress of dialogue generation research.

5.2 Humor Generation

Humor generation is another relevant area. Previous studies mainly focused on specific types of humor, such as puns [25]. However, the crosstalk is a comprehensive performance art that encompasses various forms of humor, including homophones, hyperboles, sarcasm, and more. Although we only conducted experiments over straightforward methods, we believed it would be beneficial to introduce semantic structure of humor into the models. Particularly, [3] preliminarily discussed the automatic generation of crosstalks. But it mainly focused on SMT based methods and the dataset is not released to the community. In contrast, our study offers benchmarks for numerous neural network-based approaches and introduces a large-scale, open-source dataset. Concurrently with our research, a parallel study was conducted by [22]. Similar to our work, they also built a dataset of crosstalk and benchmark. But they focused exclusively on the generation task and employed token overlap-based metrics for evaluation, which we argue is insufficient for comprehensive assessment. As a result, we have incorporated the CRS task as an additional component to provide a more objective evaluation of performance in this domain.

6 Conclusions

In this paper, we proposed an open-source Chinese crosstalk dataset *CrossDial* for both the CRG task and the CRS task, and investigated the possibility of automatic generation of entertaining dialogues in Chinese crosstalks. Through experiments, we found that special language patterns of Chinese crosstalks were difficult for current neural models. We provided two performance benchmarks and hoped that they would push forward the automatic generation of the traditional Chinese art.

References

1. Danescu-Niculescu-Mizil, C., Lee, L.: Chameleons in imagined conversations: a new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics, Portland, Oregon, USA. Association for Computational Linguistics, June 2011

2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019)
3. Du, S., Wan, X., Ye, Y.: Towards automatic generation of entertaining dialogues in Chinese crosstalks
4. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, November 2021
5. Giorgi, J., Nitski, O., Wang, B., Bader, G.: DeCLUTR: deep contrastive learning for unsupervised textual representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, August 2021
6. Kim, T., Yoo, K.M., Lee, S.G.: Self-guided contrastive learning for BERT sentence representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, August 2021
7. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, July 2020
8. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California. Association for Computational Linguistics, June 2016
9. Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., Gao, J.: Deep reinforcement learning for dialogue generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas. Association for Computational Linguistics, November 2016
10. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark. Association for Computational Linguistics, September 2017
11. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, Spain. Association for Computational Linguistics, July 2004
12. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach (2019)
13. Lowe, R., Pow, N., Serban, I., Pineau, J.: The Ubuntu dialogue corpus: a large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic. Association for Computational Linguistics, September 2015
14. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics, July 2002
15. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
16. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
17. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks (2019)

18. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models (2016)
19. Sun, Y., et al.: ERNIE: enhanced representation through knowledge integration (2019)
20. Tiedemann, J.: Parallel data, tools and interfaces in OPUS. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey. European Language Resources Association (ELRA), May 2012
21. Vaswani, A., et al.: Attention is all you need (2017)
22. Wang, B., Wu, X., Liu, X., Li, J., Tiwari, P., Xie, Q.: Can language models make fun? A case study in Chinese comical crosstalk (2022)
23. Wen, T.H., et al.: A network-based end-to-end trainable task-oriented dialogue system (2017)
24. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada. Association for Computational Linguistics, July 2017
25. Yu, Z., Zang, H., Wan, X.: Homophonic pun generation with lexically constrained rewriting. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, November 2020
26. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: i have a dog, do you have pets too? In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia. Association for Computational Linguistics, July 2018

Oral: Fundamentals of NLP

Recurrent Transformers for Long Document Understanding

Chuzhan Hao[1], Peng Zhang[1](✉), Minghui Xie[1], and Dongming Zhao[2]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{chuzhanhao,pzhang,minghuixie}@tju.edu.cn

[2] Artificial Intelligence Laboratory, China Mobile Communication Group Tianjin Co., Ltd, Tianjin, China

Abstract. Pre-trained models have been proved effective in natural language understanding. For long document understanding, the key challenges are long-range dependence and inference efficiency. Existing approaches, however, (i) usually cannot fully model the context structure and global semantics within a long document, (ii) and lack consistency assessment on common downstream tasks. To address these issues, we propose a novel Recurrent Transformers (RTrans) for long document understanding which can not only learn long contextual structure and relationships, but also be extended to diverse downstream tasks. Specifically, our model introduces recurrent transformer block to convey the token-level contextual information across segments and capture long-range dependence. The ranking strategy is utilized to aggregate the local and global information for final prediction. Experiments on diverse tasks that require understanding long document demonstrate superior and robust performance of RTrans and our approach achieves a better balance between effectiveness and efficiency.

Keywords: Recurrent mechanism · Text representation · Long document task

1 Introduction

The large scale pre-trained language models (PLMs) [1,2] have achieved great success in natural language processing tasks. Typically, these models consist of a stack of transformer layers that only encode a length-limited sequence (*e.g.*, 512 or 1024).[1] However, the real-scenario data can be arbitrarily long. Therefore, how to understand longer document effectively and efficiently becomes a key challenge [3–5].

Existing research for long document understanding has three main directions. One influential direction is to replace the full self-attention with sparse attention patterns to reduce the computation complexity to $\mathcal{O}(N)$ [5,6]. The second is to compress the sequence length. [4,7] select the key sentences by

[1] https://github.com/HAOChuzhan/RTrans.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 57–68, 2023.
https://doi.org/10.1007/978-3-031-44693-1_5

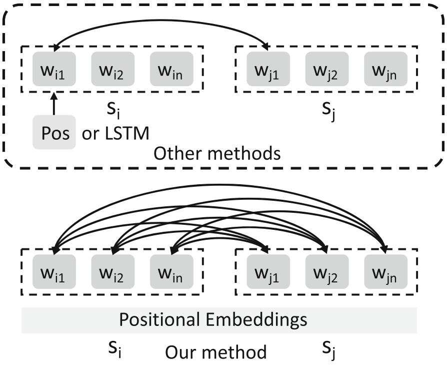


Fig. 1. s_i and s_j are adjacent text segments. w_{ij} denotes the *jth* token of the *ith* segment. Our method conveys more full token-level information across text segments compared with other models, which makes each token perceive long-range contextual information. Additionally, the recurrent modeling method is naturally suitable for sequential document structure.

feature importance and gradients. The third is to chunk a long sequence into much shorter ones with a sliding window, then build connections between the shorter sequences. Furthermore, applying recurrence to transformer has made much progress in language modeling [3,8], which demonstrates the advantage of recurrence in modeling sequential structure to some extend. Meanwhile, it is an orthogonal direction comparing to the first efficient attention approaches.

However, existing methods have still face some critical problems. First, these methods have not effectively modeled the global context structure and semantics. Previous works [9,10] usually model recurrence across text segments using recurrent neural networks (RNNs) or gates which only convey segment-level information in a single state vector (shown in Fig. 1). The receptive field of the single state vector is limited to corresponding segment length.

Second, the relative performance of most models designed for long document is unclear due to lack of consistency assessment on common tasks. Most models [3,6,8,11,12] primarily focus on autoregressive language modeling or are experimented on the different relatively short datasets. Although the recurrent modeling methods and sparse attention mechanisms [3,5,11] have superior performance in language modeling of long document, the capacity of them has not been fully demonstrated in other long document tasks. Even they are far less effective than the BERT/RoBERTa baseline in the classification task [13].

In response, we propose a novel recurrent transformers (RTrans) for efficient and effective long document understanding based on segment-level recurrence. We firstly divide the long document into overlapping/non-overlapping segments of a predefined length according to different tasks. These segments are encoded using a shared pre-trained encoder. Subsequently, to fully model the sequential structure and global semantics of document, we design a Recurrent Transformer (RT) block to convey token-level information and build connections between segments (see Sect. 3.2). The local segment representations are updated after the recurrent process, which can be applied to diverse long document tasks like Classification, QA, Reading Comprehension, Text Summarization and so on. In addition, we fully leverage the local and global representations for final answer prediction. Specifically, we take all predictions into account and choose the answer with the highest probability. Our key contributions are as follows:

- We propose a recurrent transformer block to convey the token-level history information across text segments, which can effectively model the sequential structure and global semantics of long document.
- We implement recurrent transformers for long document understanding, which breaks the limit of document length and can be easily extended to common downstream tasks.
- Extensive experiments on long document benchmarks are conducted to verify the effectiveness of our approach. Meanwhile, it achieves a good balance between effectiveness and efficiency, which also provides a reasonable reference for many practical applications.

2 Related Work

To address the limitation of encoding the full context of long document using PLMs, existing approaches are divided into three categories: 1) efficient sparse self-attention, 2) document compression, and 3) document segmentation.

Sparse strategies such as Longformer [5], Bigbird [14], Routing Transformers [11], and ClusterFormer [15] design various sparse attention mechanisms to reduce the computational complexity. Reformer [6] uses locality-sensitive hashing for content-based group attention. These models need to be trained from scratch based on new strategies and have difficulty learning the global semantics relying on only few global tokens.

Another approach is to reduce the sequence length by truncation, selection, or compressing. CogLTX [4] computes the gradients to select the key sentences. Linformer [16] applies a linear transformation to the key and value matrices to reduce the sequence length. Compressive transformers [12] and funnel transformers [17] apply additional learned compression layers to compress the sequence. However, the limited compressed space does not adequately contain valid information for some extra long documents [13].

The last approach usually divides the document into fragments and then uses a shared pre-trained model to encode each segment. Previous works [9,18] try to aggregate local representations from each segment by Pooling, LSTM, MLP, or Transformer layer over them. However, these methods are still weak at long-distance interaction and sequential modeling. In addition, some models that combine transformer architecture with recurrence [3,8] act as RNNs with states to perform language modeling based on consecutive segments, which have achieved better performance in modeling the sequential structure. However, they lack consistency assessment on diverse downstream tasks, which results in unclear relative performance among different models.

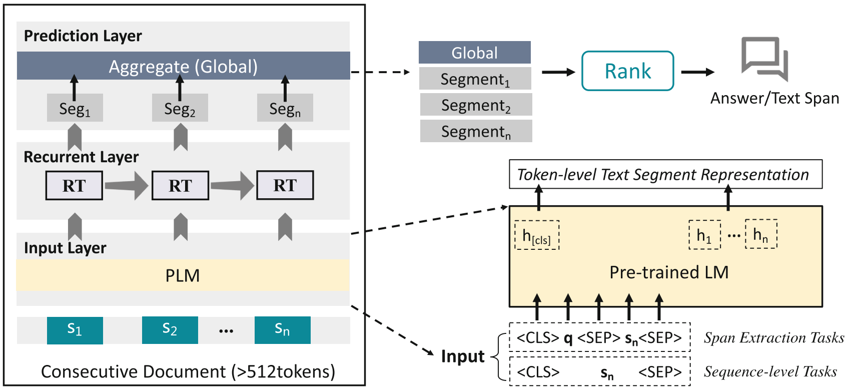


Fig. 2. The schematic diagram of our proposed RTrans. There are three major components: (1) The input layer splits D into multiple segments and encodes them using a shared PLM encoder. (2) The recurrent layer infuses token-level history information into current segment and models the sequential structure of the document. (3) The prediction layer aggregates learned multi-dimension features to solve diverse tasks.

3 Methodology

3.1 Framework Overview

To address the aforementioned limitations of existing methods, we propose the RTrans to better capture and incorporate the global contextual semantics for long document understanding. Given a consecutive long document $\mathcal{D}$ (>512 tokens), the tasks of sequence-level and span extraction aim to understand the long-range contextual information. Because most transformer-based pre-trained models have a limit of 512 for the number of tokens that they analyze at one time, we usually adopt some truncation methods or the idea of divide → embed → aggregate to balance efficiency. When we can fully establish connections between the shorter sequences, theoretically each token in the long document can receive longer distance context information.

The architecture of our proposed RTrans is illustrated in Fig. 2. Different from previous methods, we leverage the contextual representations of multiple segments from PLM and the past token-level hidden state to guide the long document understanding. To this end, the document $\mathcal{D}$ is divided into n segments $\{s_1, s_2, \cdots, s_n\}$. Sequentially, we employ the recurrent transformer block to fully convey local information across segments. The combination of local and global information and natural recurrent structure help our model to understand better global semantics, which effectively capture valid text information for various downstream tasks.

Why would our model work? Transformer-XL [8] and BRT [3] based on segment-level recurrence achieve superior performance in language modeling. This demonstrates the intrinsic advantages of the recurrent mechanism from the experiments and intuition. Based on the core idea of the above models, our proposed RTrans enhances the local segment interaction through token-level attention and positional information, and models the natural sequential document structure using recurrent mechanism. All segments share the same encoder for feature extraction without introducing too many extra parameters. Meanwhile, the large segment length allows each token to perceive long-range contextual information. Finally, the flexible settings allow us to apply our model to diverse tasks.

3.2 Recurrent Transformer Block

Figure 3 shows the architecture of the introduced *Recurrent Transformer Block*. Let S^k be the kth segment embeddings that encoded by a shared pre-trained encoder. Let H^k be the hidden embeddings from previous segment that are initialized with zeroes and contains token-level recurrent information. S^k and H^k have the same shape.

$$\mathbf{S}^k = \textsc{Plm}(\omega_{CLS}^k, \omega_1^k, \cdots, \omega_n^k, \omega_{SEP}^k) \tag{1}$$

where ω_i^k denotes the ith token of the kth segment. $S^k = \{h_{CLS}^k, h_1^k, \cdots, h_{SEP}^k\}$ denotes the representations of each token learned by Plm.

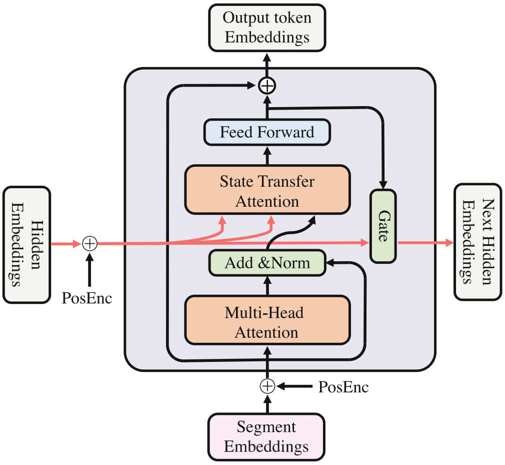


Fig. 3. The schematic diagram of Recurrent Transformer (RT) block.

On top of Plm encoder, we add the RT block to model the sequential structure. In the segment-wise encoding, the position embeddings equipped in Plm are recalculated in each segment, thereby losing the exact position of each token in the entire documentation. This positional bias may lead to inferior performance [8]. To address this problem, we assign the Rotary Position Embedding [19] to each segment as a complementary feature, indicating its relative position in the document.

We refer to the GRU [20] and Transformer decoder [21] for designing recurrent transformers. Similarly, the RT block receives two tensors as inputs which are no longer a single vector, but the N tokens embeddings. The outputs of RT block are also two tensors. O^k is used to be as a local representation of segment, and the other H^k is conveyed to the next segment as a hidden state. The whole process is computed as follows:

$$\mathbf{X}^k = \textsc{Ffn}(\textsc{Sta}(\mathbf{H_p^{k-1}}, \textsc{Ln}((\textsc{Mha}(\mathbf{S_p^k}))) \tag{2}$$

$$\mathbf{O^k} = \mathbf{S_p^k} + \mathbf{X^k} \tag{3}$$

$$\mathbf{H}^k = \textsc{Gate}(\mathbf{H}_p^{k-1}, \mathbf{X}^k) \tag{4}$$

Where S_p^* and H_p^* denote state vectors with positional information. $\textsc{Mha}, \textsc{Sta}$ are the layers of Multi-Head and Cross Attention with mask mechanism respectively. The implement of $\textsc{Gate}$ is similar to LSTM gate [3].

To make our network have the sequence information in both directions forward (past to future) or backward (future to past), we improve our approach to be bidirectional. Similar to the forward process, this backward process chunks the long document into multiple shorter segments from back to front, and then we use RT block to model the document structure in reverse as well.

3.3 Prediction Layer

After obtaining the segment representations with history information, we use a naive attention mechanism to aggregate the local segment information.

$$\mathbf{a} = \text{softmax}(\mathcal{F}_o([\mathrm{O}_n^{f1}; \cdots ; \mathrm{O}_n^{fk}])) \tag{5}$$

$$\mathbf{G}_{\mathcal{D}} = \sum_{t=1}^{k} \mathbf{a}^t \mathrm{O}_n^{ft} \tag{6}$$

where O_n^{fk} is the last hidden representation from the forward segment representations O^{fk}. $\mathrm{G}_{\mathcal{D}}$ is the global feature of long document $\mathcal{D}$. $\mathcal{F}_o$ denotes a linear fully connected layer. Therefore, we obtain the local and global representations. For the bidirectional modeling, we also use attention mechanism to aggregate reverse representations. The final global representation $\mathrm{G}_{\mathcal{D}}$ is then obtained by concatenating the forward $\mathrm{G}_{\mathcal{D}}^f$ and backward $\mathrm{G}_{\mathcal{D}}^b$ document representations. This can better capture bidirectional semantic dependencies.

For document classification task, the contextual global representation $\mathrm{G}_{\mathcal{D}}$ is used to train and infer. We use appropriate loss functions for different classification tasks. Following [5], we use BCE/CE loss for classification tasks. Furthermore, we train the answer extraction task via supervised learning. The predictions of the answer span are based on multiple text segments. We sort the start and end positions of different segments by their logits, and select the i^*-th and j^*-th tokens with the highest logit as our predicted answer start and end, respectively. For the long document extractive summarization task, the training objective of the model is to minimize the binary cross entropy loss given the predictions and ground truth sentence labels.

4 Experiments

4.1 Experimental Settings

Datasets. To evaluate the effectiveness of RTrans, we conduct experiments on various long document datasets of different domains: 1) Hyperpartisan [22] contains news articles which are manually labelled as hyperpartisan (taking an extreme left or right standpoint) or not. 2) 20NewsGroups [23] contains newsgroups posts which are categorised into 20 topics. 3) IMDB [24] is a widely used dataset for movie review rating prediction. We modify 4) CMU Book Summary [25] and 5) EURLEX-57K [26] following prior work [13]. We show the statistics of all datasets in Table 2.

Implementation Details. For the PLM encoder, we use the "bert-base-uncased" version with the hidden size of 768 and fine-tune it for document classification. For other downstream tasks, we use the same PLM encoder as the baseline model. Across all datasets, we use Adam optimizer with a learning rate of {5e−5, 3e−5, 1e−5, 5e−3} for one run of each model and pick the best performing learning rate. The segment length needs to be selected by experiments from {64, 128, 256, 384, 512}. Other model settings (e.g., positional encoding methods, gate configuration and PLM scale and type) depend on experimental results.

Baselines and Metrics. We compare our model with prior competitive methods in diverse tasks. (1) Long Document Classification: BERT [1] (*+Head* uses the first 512 tokens from the document instance. *+Slide* divides the document into multiple segments in a fixed window length and aggregates the $<CLS>$ vectors of all segments with naive attention), ToBERT [9], CogLTX [4], Longformer [5], Bigbird [14], Hi-Transformer [18], etc. We report the metrics of accuracy, macro-F_1 and micro-F_1. (2) Machine Reading Comprehension: BERT [1] and BERT-RCM [10]. The evaluation metric is macro-average word-level F_1 score. (3) Text Summarization: Longformer-Ext [5], GRETEL, etc. We report the unigram (ROUGE-1), bigram F1 (ROUGE-2), and the longest common subsequence (ROUGE-L) between the generated summary and gold summary.

4.2 Main Results

Document Classification. In many practical application scenarios, many problems can be translated into a classification task. Thus, we firstly conduct experiments in various long document classification tasks, including binary, multi-class and multi-label classification. To further capture bidirectional semantic dependencies, we propose a Bi-directional RTrans, referred to as Bi-RTrans. Table 1 shows that our approaches outperform common strong baselines designed for long document. Overall, the key conclusions are that (1) the simple baselines

BERT+ can achieve approximate performance compared with the more sophisticated models (ToBERT, CogLTX, Longformer) and (2) receiving longer *global information* and *modeling the sequential structure* of the long document are critical for long document classification tasks. Specifically, for the datasets of Hyperpartisan, 20NewsGroups, Book Summary and ECtHR, our approaches achieve the best performance. We can observe that the performance of the BERT-Head model deteriorates ($-1.23\% \rightarrow -1.38\% \rightarrow -5.00\% \rightarrow -9.43\%$) as the average length of datasets increases (20News→Book→Hyper→ECtHR), which demonstrates that truncating long document results in the loss of large amounts of valid information. In addition, BERT-Slide performs better on the EURLEX datasets. This is because the context of these datasets lacks correlation and each text segment is relatively independent, so sequential modeling is not beneficial. [26] also shows that processing the first two sections only (header and recitals) results in almost the same performance as the full documents in EURLEX, which explains why the BERT-Head model has superior performance.

Meanwhile, we conduct experiments to observe the relative efficiency and effectiveness of these existing models in Hyperpartisan. Compared to our RTrans, Longformer takes about 15x, 5.7x more time for training and inference while CogLTX takes even longer. Overall, our proposed RTrans not only achieves the approximate speed and memory usage with BERT+ model, but also achieves the best performance compared with the more sophisticated models.

Table 1. Evaluation of performance metrics on documents with more than 512 tokens in the test set for all datasets. These strong baselines belong to the three categories mentioned in related work respectively. The average accuracy (%) over five runs is reported for Hyperpartisan and 20NewsGroups while the average micro-F$_1$ (%) is used for the other datasets. Partial results are from [13]. The highest value per column is in bold and the second highest value is underlined.

Methods	#Param	Hyperpartisan	20News Groups	ECtHR	Book Summary	Paired Book Summary	EURLEX	Inverted EURLEX
BERT+Head	110M	91.50	86.36	68.98	62.12	54.53	66.76	62.88
BERT+Slide	110M	92.50	83.96	77.55	63.02	60.89	**67.17**	**64.48**
ToBERT	115M	86.50	–	–	61.38	58.17	61.85	59.50
CogLTX	–	91.91	86.07	–	60.71	55.74	61.95	63.00
Longformer	149M	94.58	85.50	77.81	59.66	58.85	44.66	47.00
RTrans	117M	**96.50**	87.14	78.26	62.64	61.24	65.84	63.60
Bi-RTrans	124M	95.00	**87.59**	**78.41**	**63.50**	**61.42**	66.45	63.39

Sentiment Analysis. The shorter texts usually have the same sentiment information in most sentiment analysis datasets, which can be easily understood by only partial key text. However, there is often unstable sentiment information in longer sentiment analysis datasets. We find that Longformer and Bigbird have worse performance than other methods in Table 3. This is because the sparse attention mechanisms used in them usually have fewer global attention tokens, which makes it difficult to capture global context information. Our approach

Table 2. Statistics of datasets. #Avg denotes the average token count obtained via the BERT base/large (uncased) tokenizer in training dataset. % Long denotes the percentage of documents with more than 512 tokens.

Dataset	Task	#Avg	%Long
Hyperpartisan	TC	737	52.71
20NewsGroups		375	14.66
IMDB		431	29.37
ECtHR	MLC	2,145	80.45
Book Summary		572	38.65
–Paired		1,145	75.62
EURLEX-57K		710	51.45
–Inverted			
TriviaQA (wiki)	MRC	2,630	92.38
PubMed-Long	TS	3,235	96.62

Table 3. Evaluation of performance metrics in the test set of sentiment analysis dataset IMDB. Partial results are from [18].

Methods	IMDB	
	Accuracy	Macro-F$_1$
BERT-Head	52.13 ± 0.59	46.34 ± 0.33
BERT-Slide	53.07 ± 0.35	47.74 ± 0.24
Longformer	52.33 ± 0.40	43.51 ± 0.42
Bigbird	52.87 ± 0.51	43.79 ± 0.50
HIBERT	52.96 ± 0.46	43.84 ± 0.46
Hi-Transformer	53.78 ± 0.49	44.54 ± 0.47
RTrans	**55.75** ± 0.43	**50.81** ± 0.42

achieves the best performance, which demonstrates that the recurrent modeling with RT block is effective to capture unstable sentiment information.

Reading Comprehension and Text Summarization. To further explore the generalization of our framework, we migrate RTrans to span extraction task (e.g. Machine Reading Comprehension). Similarly, we conduct experiments on an extra long document dataset, TriviaQA. Follow previous work [10], we compare the Gated and LSTM recurrent mechanisms to our recurrent transformer block. The experimental results in Table 4 show that our proposed RTrans achieves consistent performance improvement (+0.4%, +1.0%), which also demonstrates that sequential modeling and conveying token-level information of segment are beneficial for long document understanding. Furthermore, to further explore the capability of our framework of understanding the sentence-level information in the long document, we conduct experiments in extractive summarization dataset, PubMed. We also introduce our RT block to model the correlations among text blocks. The results are shown in Table 5.

4.3 Ablation Studies

We further conduct specific experiments to investigate the effectiveness, necessity and applicability of our proposed RTrans.

Impact of Model Components. To demonstrate the effectiveness of our recurrent modeling, we compare it with other aggregation methods, e.g., MLP, LSTM, Gate and Transformer. Table 6 presents the results on different methods of building connections. Our approach has improved performance across different datasets and metrics. In the 20NewsGroups dataset, our RT block achieves

Table 4. Replace Recurrent Chunking Mechanisms in [10] with our proposed RT block and then evaluate the performance on the TriviaQA dataset.

Methods	TriviaQA
	F$_1$ (%)
BERT-Large	61.3
Sent-Selector	59.8
BERT-RCM	
–Gated recurrence	62.9
–LSTM recurrence	62.3
RTrans	**63.3**

Table 5. ROUGE F$_1$ results of different models on PubMed-Long.

Methods	PubMed-Long		
	R-1	R-2	R-L
BERTSum	41.09	15.51	36.85
Longformer-Ext	43.75	17.37	39.71
Reformer-Ext	42.32	15.91	38.26
BERTSum+SW	42.50	19.67	42.49
GBT-EXTSUM	46.87	20.19	42.68
GRETEL	48.20	21.20	43.16
RTrans	**48.96**	**21.32**	**43.57**

Table 6. Ablation study of our model components (replacing RT with other methods each time), using the 20NewsGroups test set and the TriviaQA validation set.

Framework	20NewsGroups	TriviaQA
	Accuracy	F$_1$
+ RT (ours)	**85.79**	**63.3**
(a) r/ Gate	85.61	62.9
(b) r/ LSTM	84.71	62.3
(c) r/ MLP	84.47	–
(d) r/ Transformer	85.52	–

Table 7. Performance on different sequence length of different models, using the ECtHR test set.

Methods	ECtHR		
	(512,1024]	(1024,2048]	(2048,+∞)
Number	175	282	419
BERT+Head	73.78	70.35	66.40
Longformer	78.80	78.98	76.66
Bigbird	78.49	78.56	76.23
ClusterFormer	74.52	74.34	73.67
RTrans	**81.13**	**79.59**	**76.86**

better results (+0.27%, +1.08%) over Transformer and LSTM, which suggests that our approach has advantages in modeling sequential structure and capturing global semantic information.

Impact of the Segment Length and Max Length. Intuitively, if the model processes the longer document content, we can get more adequate global information. The shorter segment length means more accurate text encoding. While the former will result in more resource consumption and slower inference speed, the latter will results in text fragmentation. Therefore, it is important to find a balanced max length and segment length. Experimental results shows that the optimal segment length is *dataset dependent* and the optimal max length is *much more than 512*. Therefore, the limited compress space is difficult to contain the global semantic information for extra long documents. For the different sequence length ranges, the experimental results in Table 7 show that our approach achieves robust performance. Compared with BERT-Head model, our model exhibits a big boost (about 10%) in different length ranges. Meanwhile, it also has a significant improvement over Longformer. All in all, our method can be well adapted to different sequence lengths.

5 Conclusions

In this paper, we propose recurrent transformers for long document understanding, which can learn the long-range document structure and global semantics and be extended to diverse downstream tasks. Experimental results demonstrate the superiority of RTrans over other models designed for long document. Furthermore, our approach also achieves a balance of effectiveness and efficiency. Therefore, it has a strong application prospect. In future work, we plan to apply our model to more tasks, and further investigate the adaptability of our model to realize rapid distributed deployment and establish robust performance.

References

1. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, June 2019, pp. 4171–4186. Association for Computational Linguistics (2019)
2. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692 (2019)
3. Hutchins, D., Schlag, I., Wu, Y., Dyer, E., Neyshabur, B.: Block-recurrent transformers, arXiv preprint arXiv:2203.07852 (2022)
4. Ding, M., Zhou, C., Yang, H., Tang, J.: CogLTX: applying BERT to long texts. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 12 792–12 804. Curran Associates Inc (2020)
5. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer, arXiv preprint arXiv:2004.05150 (2020)
6. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: International Conference on Learning Representations (2020)
7. Fiok, K., et al.: Text guide: improving the quality of long text classification by a text selection method based on feature importance. IEEE Access **9**, 105 439–105 450 (2021)
8. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R.: Transformer-XL: attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy, July 2019, pp. 2978–2988. Association for Computational Linguistics (2019)
9. Pappagari, R., Żelasko, P., Villalba, J., Carmiel, Y., Dehak, N.: Hierarchical transformers for long document classification. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 838–844 (2019)
10. Gong, H., Shen, Y., Yu, D., Chen, J., Yu, D.: Recurrent chunking mechanisms for long-text machine reading comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, July 2020. Online, pp. 6751–6761 (2020)
11. Roy, A., Saffar, M., Vaswani, A., Grangier, D.: Efficient content-based sparse attention with routing transformers. Trans. Assoc. Comput. Linguist. **9**, 53–68 (2021)

12. Rae, J.W., Potapenko, A., Jayakumar, S.M., Hillier, C., Lillicrap, T.P.: Compressive transformers for long-range sequence modelling. In: International Conference on Learning Representations (2020)
13. Park, H.H., Vyas, Y., Shah, K.: Efficient classification of long documents using transformers, arXiv preprint arXiv:2203.11258 (2022)
14. Zaheer, M., et al.: Big bird: transformers for longer sequences. In: Advances in Neural Information Processing Systems, vol. 33, pp. 17 283–17 297 (2020)
15. Wang, N., et al.: ClusterFormer: neural clustering attention for efficient and effective transformer. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 2022, pp. 2390–2402. Association for Computational Linguistics (2022)
16. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: self-attention with linear complexity, arXiv preprint arXiv:2006.04768 (2020)
17. Dai, Z., Lai, G., Yang, Y., Le, Q.: Funnel-transformer: filtering out sequential redundancy for efficient language processing. Adv. Neural. Inf. Process. Syst. **33**, 4271–4282 (2020)
18. Wu, C., Wu, F., Qi, T., Huang, Y.: Hi-transformer: hierarchical interactive transformer for efficient and effective long document modeling. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 848–853. Online: Association for Computational Linguistics, August 2021
19. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: enhanced transformer with rotary position embedding (2021)
20. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 2014, pp. 1724–1734. Association for Computational Linguistics (2014)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
22. Kiesel, J., et al.: SemEval-2019 task 4: hyperpartisan news detection. In: Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, Minnesota, USA, June 2019, pp. 829–839. Association for Computational Linguistics (2019)
23. Lang, K.: NewsWeeder: learning to filter netnews. In: Machine Learning Proceedings, pp. 331–339. Elsevier (1995)
24. Diao, Q., Qiu, M., Wu, C.-Y., Smola, A.J., Jiang, J., Wang, C.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 193–202 (2014)
25. Bamman, D., Smith, N.A.: New alignment methods for discriminative book summarization, arXiv preprint arXiv:1305.1319 (2013)
26. Chalkidis, I., Fergadiotis, E., Malakasiotis, P., Androutsopoulos, I.: Large-scale multi-label text classification on EU legislation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019, pp. 6314–6322. Association for Computational Linguistics (2019)

SCA-CLS: A New Semantic-Context-Aware Framework for Community-Oriented Lexical Simplification

Rongying Li[1], Wenxiu Xie[2], John Lee[3], and Tianyong Hao[4](✉)

[1] School of Artificial Intelligence, South China Normal University, Guangzhou, China
Kitty-LRY@outlook.com

[2] Department of Computer Science, City University of Hong Kong, Hong Kong, China
vasiliky@outlook.com

[3] Department of Linguistics and Translation, City University of Hong Kong, Hong Kong, China
jsylee@cityu.edu.hk

[4] School of Computer Science, South China Normal University, Guangzhou, China
haoty@m.scnu.edu.cn

Abstract. Community-oriented lexical simplification aims to transform complex words within a sentence into semantically consistent but simple substitute words from a community-specific vocabulary. Most state-of-the-art contextual word embedding models generate substitutes by extracting contextual information of complex words. Although these models take context into account, they fail to capture rich semantics of complex words with polysemy, resulting in many spurious and semantically non-equivalent candidates. Thus, this paper proposes a novel Semantic-Context-Aware framework for Community-oriented Lexical Simplification (SCA-CLS), which integrates gloss (sense definition) into BERT to identify the actual sense of the complex word (especially for polysemy) in current context and ranks substitutes by proposed gloss similarity. In addition, a new complexity feature is proposed to enhance substitute ranking. Experiment results on Wikipedia dataset show that SCA-CLS outperforms the state-of-the-art Merge-Sort model on both substitute generation and ranking tasks, indicating its effectiveness for community-oriented lexical simplification.

Keywords: Lexical simplification · BERT · Gloss · Ranking · Semantic

1 Introduction

Lexical simplification (LS) is a task to replace complex words in a sentence with words that are easier to read (or understand) without changing the semantic meaning of the original sentence. LS is an effective way to reduce the difficulty of reading, especially for the population with dyslexia, aphasia and poor literacy [1]. Some studies have shown that as long as readers have a sufficient number of familiar words (relatively simple words), even if the grammatical structures used in a sentence are confusing, those familiar words can often help them comprehend the sentence [2, 3]. LS has been applied to many practical scenarios, such as a reading aid for end users and a preprocessing step for other natural language processing tasks [4].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 69–81, 2023.
https://doi.org/10.1007/978-3-031-44693-1_6

Traditional LS methods return a unified simple word for all readers, regardless of diverse language and education levels of readers, which lead to ineffective simplification (simplified words are still incomprehensible for certain readers). Thus, Community-oriented Lexical Simplification (CLS) is proposed to solve this problem [5]. CLS provides users with substitutes from a community list, where simple words match the cognition levels of targeted community users. For instance, the Hong Kong Education Bureau has formulated a vocabulary list for primary school students, hereinafter called "EDB list", which contains the vocabulary required for basic primary education [6]. The CLS restricts the candidates with a community vocabulary list and filters out those generated candidates not in the list. For example, the generated candidate of the target word "*confrontation*" for sentence "*A final violent confrontation with police took place at Glenrowan.*" is "*combat*". However, "*combat*" is discarded if not on the EDB list. This restriction of substitutes makes CLS more challenging than the ordinary LS task.

Contextual word embedding models such as ELMO [7] and BERT [8] are usually pre-trained on large corpora, and they can learn different embedding representations for the same word according to specific context. Qiang et al. [9] achieved state-of-the-art results in LS task by applying BERT's learning strategy NSP (Next Sentence Prediction) to capture sentence contextual information. However, contextual word embedding models may produce substitutes that fit the context but are semantically dissimilar or opposite to target words, although they take context into account [9].

This paper proposes a Semantic-Context-Aware framework for Community-oriented Lexical Simplification (SCA-CLS), which considers both contextual and actual word sense information in substitute generation and ranking subtasks. SCA-CLS integrates gloss (sense definition) from WordNet into the candidate prediction of BERT. Lexical resources contribute to improving model performance by providing additional semantic information to ensure that generated candidates are semantically similar to target words in addition to being contextually suitable. The major contributions of the paper lie in:

1) A new Semantic-Context-Aware framework SCA-CLS is proposed to utilize both semantic (actual word sense) and contextual information in generating substitutes by introducing gloss into BERT and adopting a new *5-gram* strategy to optimize NSP.
2) A gloss similarity metric and a complexity feature are proposed to rank generated candidates, where the metric is for semantic similarity calculation between target words and substitutes, and the feature is for complexity measurement of substitutes.
3) Experiment results show that SCA-CLS outperforms baseline methods including the state-of-the-art Merge-Sort model [10], indicating its effectiveness in the CLS task.

2 Related Work

LS typically consists of three main steps: substitute generation, substitute filtering and substitute ranking. Each step is closely interlinked and plays a major role in LS [11]. Existing LS methods mainly focus on substitute generation and ranking. The earliest research mainly used the lexicon extraction rules for LS [12]. Some studies used synonyms, hypernyms and hyponyms extracted from WordNet as candidates for complex words [13, 14]. Although WordNet has proven useful for LS, Shardlow et al. [15] showed that using WordNet synonyms alone limits the potential of simplification, as WordNet can cover neither all complex words nor all potential substitutes.

To avoid depending on semantic lexicon or parallel corpora, Glavas et al. [16] proffered an unsupervised method to train word embedding models, so that each word in the corpus can be represented by a unique embedding. However, such word embedding models cannot effectively simplify words with multiple semantics because they do not consider contextual information. Melamud et al. [17] proposed a Context2vec model that used Bi-LSTM (Bi-directional Long Short-Term Memory) to obtain a neural representation of the complete sentence context, making it excellent in LS and WSD (Word Sense Disambiguation). Peters et al. [7] published ELMO (Embeddings from Language Models), which used forward and reverse neural networks to capture context information and provided a multidimensional vector representing semantics for each word. Qing et al. [9] proposed LSBert, which applied NSP to capture contextual information of target words and used MLM (Masked Language Model) to make predictions. Since the complexity of polysemy is closely related to context, contextual word embedding models considering entire contextual information can help determine the semantics of polysemy in current context to some extent, but they are still not accurate enough.

Many studies leveraged external language knowledge to improve the performance of deep learning models. LexSubCon is a LS framework presented by Michalopoulos et al. [18], which proposes a new mix-up embedding strategy to integrate contextual information and knowledge from structured lexical resources. Yap et al. [19] fine-tuned BERT using annotated sentences from SemCor [20] and sense definitions from WordNet, treating WSD as a relevance ranking task.

Recently, researchers have shown interest in CLS which predicts candidates are within community users' cognitive level [21, 22]. In practice, users have different familiarity and cognition of words for the different personal knowledge levels and language backgrounds. Hao et al. [5] proposed a semantic-context ranking method that selected substitutes from the EDB list and ranked them using WordNet and Google 1T 5-gram corpus. Song et al. [23] further improved Hao et al.'s method by using a new context-aware method based on combining Word2vec and N-gram for semantic and context ranking. However, these methods mainly focused on substitute ranking, resulting in low efficiency of substitute generation (generating many false candidates) and a small improvement in ranking performance. Song et al. [24] proposed a hybrid model that employed a weighted average method to merge the candidates generated by Context2vec and context-aware models. Li et al. [10] developed a context-driven Merge-Sort model that integrated and reranked the candidates generated from BERT and Context2vec models by a new rank-inverse weighting method. However, producing a large number of candidates does not mean an improvement in quality.

3 The SCA-CLS Framework

For CLS, the generated substitutes must meet the cognitive level of specific groups and be semantically similar to the target word and suitable for the context. To this end, we propose a SCA-CLS framework, shown in Fig. 1, which generates substitutes through the proposed hybrid embedding strategy, filters substandard substitutes based on semantic similarity, word frequency, and community vocabulary, and ranks substitutes by the proposed semantic and complexity-based ranking score.

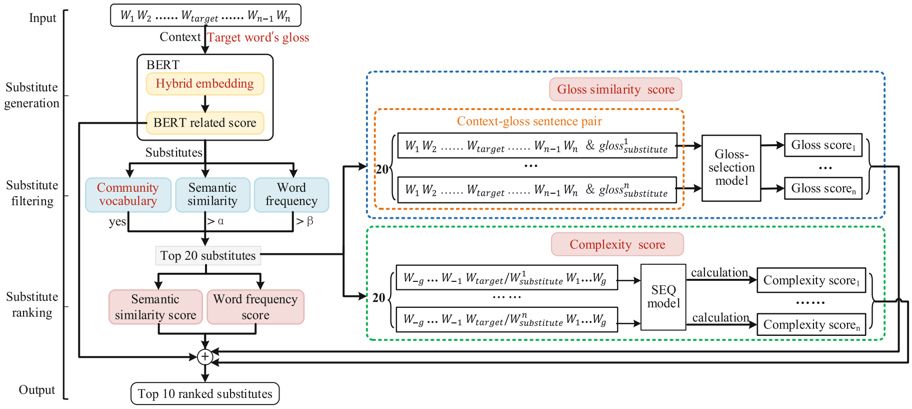

Input
Substitute generation
Substitute filtering
Substitute ranking
Output
$W_1\ W_2 \ldots\ldots W_{target} \ldots\ldots W_{n-1}\ W_n$
Context
Target word's gloss
BERT
Hybrid embedding
BERT related score
Substitutes
Community vocabulary
Semantic similarity
Word frequency
yes
> α
> β
Top 20 substitutes
Semantic similarity score
Word frequency score
Top 10 ranked substitutes
Gloss similarity score
Context-gloss sentence pair
20
$W_1\ W_2 \ldots\ldots W_{target} \ldots\ldots W_{n-1}\ W_n$ & $gloss^1_{substitute}$
$W_1\ W_2 \ldots\ldots W_{target} \ldots\ldots W_{n-1}\ W_n$ & $gloss^n_{substitute}$
Gloss-selection model
Gloss score$_1$
Gloss score$_n$
Complexity score
20
$W_{-g} \ldots W_{-1}\ W_{target}/W^1_{substitute}\ W_1 \ldots W_g$
$W_{-g} \ldots W_{-1}\ W_{target}/W^n_{substitute}\ W_1 \ldots W_g$
SEQ model
calculation
calculation
Complexity score$_1$
Complexity score$_n$

Fig. 1. The SCA-CLS framework for community-oriented lexical simplification.

3.1 Gloss Integration and Gloss Similarity

For the sentence S and the target word w, CLS needs to generate a set of appropriate candidates based on context and the semantics of target word w. SCA-CLS generates substitutes using a pre-trained language model BERT. Directly using BERT to predict masked target words may produce substitutes that fit the current context but are semantically different from target words [9]. Thus, we introduce glosses (sense definition) into BERT to capture the actual word sense of the target word in current context. To select the most appropriate glosses for target words, we utilize a pre-trained BERT-based gloss-selection model published by Yap et al. [19], which yielded SOTA results in WSD task. The gloss-selection model provides a list of potential glosses from WordNet for each target word, and calculates the correlation score for each gloss based on sentence contextual information and target word itself. The gloss with the highest score means that its sense is closest to the semantics of the target word in the current context. Since glosses are sentences (word sequences), we utilize BERT's NSP strategy to generate substitutes. We join the best gloss S_{gloss} of the target word and the target masked sentence S' into a sentence pair (S_{gloss}, S') as input to BERT. For instance, the best gloss of the target word "*prosperous*" for sentence "*Nuremberg is a prosperous city*" is "*very lively and profitable*", so the sentence pair (S_{gloss}, S') is "[CLS] *very lively and profitable* [SEP] *Nuremberg is a* [MASK] *city* [SEP]", where "[CLS]" and "[SEP]" tokens are added to help BERT accept statement-level input. We return the target word's embedding of the last output layer for subsequent hybrid embeddings.

Gloss is a short explanation for difficult or obscure words. If two words are semantically similar, their glosses are highly similar or at least contain identical or similar keywords. Therefore, we propose to rank substitutes by the gloss similarity between target words and substitutes. The gloss-selection model is based on BERT, which takes context-gloss pairs as input and calculates the relevance score between the target word in context and its glosses from WordNet. First, we replace the target word in the sentence S with generated substitutes and use the gloss-selection model to find the best gloss for each substitute, producing a list of substitutes' best glosses $G = \{g_1, g_2, \ldots, g_n\}$. Then, each gloss in G is used to combine with sentence S to form a new context-gloss sentence pair ([CLS] S [SEP] g_n [SEP]) as input to obtain a relevance score that reflects how similar the gloss is between the target word and the substitute. The higher the score, the closer between the target word and the candidate.

3.2 Complexity Measurement

Complex Word Identification (CWI) task detects which words in a sentence need to be simplified. Generally speaking, words with lower complexity are less difficult to read and more familiar to people. Here, we apply CWI to rank substitutes. We measure the complexity of substitutes using the sequence labelling (SEQ) model published by Gooding et al. [25], which treats CWI as a sequence labelling task. Instead of using extensive feature engineering, the SEQ model returns a binary complexity score for each word in the sentence by utilizing word embeddings and sentence context. The original sentence and the new sentence with the target word replaced by generated substitutes are respectively entered into the SEQ model to obtain a complexity score for each word. Context impacts the perceived complexity of text [25]. Ideally, a suitable candidate can make context simpler, so we use a symmetrical window of size g ($W = w_{-g}, \dots, w_{-1}, w, w_1, \dots, w_g$) around the target word w to measure the change in context complexity, where g is set to 5. The complexity change r_{com} is calculated by Eq. (1), where c_w denotes the complexity difference between target words and substitutes, and c_{w_i} is the complexity difference of the i^{th} word within W before and after the candidate replacement. A high complexity change score means that substitutes have a more positive effect on reducing complexity of sentences and target words.

$$r_{com} = c_w + \frac{1}{2g}\left(\sum_{i=-g}^{i=-1} c_{w_i} + \sum_{i=1}^{i=g} c_{w_i}\right) \tag{1}$$

3.3 NSP Optimization for Hybrid Embedding

BERT introduces two tasks in the pre-training process, masked language model (MLM) and next-sentence prediction (NSP), the former based on single sentences and the latter based on sentence pairs [8]. Qiang et al. [9] showed that using NSP is more effective for LS than using MLM. As the sentence pair (S, S') connected by the original sentence S and the target masked sentence S' is input into BERT, contextual information of target words is used twice, which leads to excessive interference of contextual information. We propose a new *5-gram* strategy to optimize NSP by using a symmetrical window of size g ($W = w_{-g}, \dots, w_{-1}, w, w_1, \dots, w_g$) around the target word w as context, where g is set to 5. In order to capture the effective contextual information and reduce the interference of distant contextual information, we mask words outside the window range in sentence S. After that, we input the resulting new sentence pair (S, S') into BERT to obtain an output embedding of each token produced by the output layer.

In order to incorporate more semantic information into contextual embeddings, we apply a hybrid embedding strategy to the target word's embedding by mixing the output embedding X_{target} obtained above with the output embedding X_{gloss} generated by BERT in Sect. 3.1. As shown in Eq. (2), X'_{target} is a new hybrid embedding of the target word and a parameter λ is set to maximize the positive impact of the contextual and semantic information contained in target word embeddings. Finally, as shown in Fig. 2, the output layer takes the hidden states of the "[CLS]" token as input to predict substitutes and calculate the relevance score.

$$X'_{target} = \lambda X_{target} + (1 - \lambda)X_{gloss} \tag{2}$$

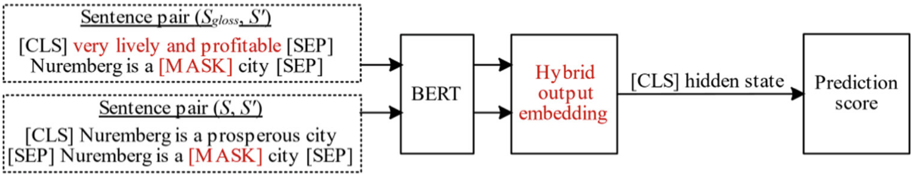

Sentence pair (S_{gloss}, S')
[CLS] very lively and profitable [SEP] Nuremberg is a [MASK] city [SEP]
Sentence pair (S, S')
[CLS] Nuremberg is a prosperous city [SEP] Nuremberg is a [MASK] city [SEP]
BERT
Hybrid output embedding
[CLS] hidden state
Prediction score

Fig. 2. Substitute generation of SCA-CLS for the target word prediction. Two sequences consisting of sentence pair (S, S') and sentence pair (S_{gloss}, S') are entered into BERT, respectively.

3.4 Substitute filtering and ranking

Substitution filtering aims to remove unreasonable candidates, thereby improving the effectiveness of subsequent substitute ranking. To obtain high-quality substitutes, three restrictions are added to filter generated candidates. A path-based WordNet method based on the shortest path between substitutes and target words in WordNet directed graph is applied to compute semantic similarity, and a semantic similarity threshold ($\alpha = 0.1$) is set to discard substitutes with low semantic similarity to target words. As words that are used more frequently tend to be considered simpler, word frequency is also adopted as a simple and effective strategy for substitute selection. We use the Zipf scale created from the SUBTLEX lists [26], where word frequencies are verified to be correlated with human judgments on simplicity [27], and set a frequency threshold $\beta = 3$ to filter out candidates with low frequencies.

In CLS, substitutes must be selected from a simple vocabulary list that meets the needs of the specific community. Thus, we use the EDB list for CLS. We utilize porter stemming tool and morphy method in NLTK to filter out those substitutes whose stemming or lemmatization is not in the EDB list. We select the top 20 substitutes generated by SCA-CLS as initial candidates, excluding morphological derivatives of target words. Then the prediction scores are mapped to [0, 1] interval using min-max normalization.

Substitute ranking aims to rank the most appropriate substitutes at the top of the substitute list to find the best substitute (the top one) for the target word. We employ four features to rank substitutes. As shown in Eq. (3),f_r denotes the final ranking score of the substitute and r_b represents the substitute score returned by BERT in Sect. 3.3. We use the average value of gloss similarity score r_g calculated by the gloss-selection model and the semantic score r_w calculated by the path-based WordNet method as the semantic-based ranking score. Word frequency reflects the complexity of words to some extent, so we use the average of word frequency score r_f from the Zipf scale [26] and the complexity score r_c calculated by the SEQ model as the complexity-based ranking score. These features inspect the applicability of substitutes from different aspects. We sum the scores of four features and the substitute score provided by BERT. Finally, top 10 substitutes are selected for evaluation.

$$f\_r = r_b + \frac{1}{2} \times (r_g + r_w + r_c + r_f) \tag{3}$$

4 Evaluation and Results

4.1 Dataset and baselines

SemEval Dataset: A standard dataset from SemEval 2007 is used as training data, involving lexical samples of nouns, verbs, adjectives and adverbs [28]. The dataset includes 295 sentences extracted from the English Internet Corpus. To ensure variety, each sense of the target word with the same part-of-speech tag has at least one instance.

Wikipedia Dataset: It contains 500 sentences manually annotated by 50 independent annotators [29]. We keep 249 sentences whose target words are not in the EDB list, but their gold answers are in the list, as **Dataset A**. Then we further remove sentences whose annotation agreements of gold candidates are below 20%, i.e., at least 10 agreements of the gold candidates from the 50 annotators, as **Dataset B**.

Baselines: We implement Context2vec, LSBert, LexSubCon and the state-of-the-art context-driven Merge-sort model as baselines. We also adopt the semantic-context ranking method [5] and the context-aware model [23] as baselines, both of which mainly focus on substitute ranking, and use their reported results for comparison.

4.2 Evaluation Metrics

The following 4 commonly used metrics are applied for performance evaluation.

Accuracy @N (A@N). The A@N is calculated as the number of correct matches (one of the generated top N candidates is in the gold set) divided by the total number of sentences. N is set from 1 to 10 to show gold set coverage of top 1 to top 10 candidates.

T is the set of test items with at least one candidate from annotators. For each $i \in T$, h_i is the gold substitute set, a_i is the set of generated candidates by model, and c_i is the set of correct candidates (a_i that in h_i). For each unique type (res) in h_i, there is an associated frequency ($freq_{res}$) indicating the number of times it has been annotated, and m_i is the best gold substitute candidate(s) with maximum frequency value in h_i.

Best Accuracy (A$_{\textbf{Best}}$). The first guess with the highest score is considered the best guess among candidates returned by system. If the first guess matches m_i, the candidate is correct and optimal for a target word. A$_{\text{Best}}$ is calculated as the number of correct matches divided by the total number of sentences, shown in Eq. (4).

Oot Accuracy ($\mathrm{A_{Oot}}$). For the 10 guesses generated by the system, if one of the guesses matches m_i, the guess is considered as correct. The $\mathrm{A_{Oot}}$ is calculated as the number of correct matches divided by the total number of sentences, as in Eq. (5).

Frequency Measure. Assign the associated frequency ($freq_{res}$) to each generated candidate in a_i: if the candidate is not in gold set h_i, $freq_{res}$ is 1, otherwise is the frequency of the gold candidate that it matches. A_i refers to the sum of the frequencies of all candidate words in a_i. H_i is the sum of the frequencies of all candidate words in h_i. Frequency Precision (**P**) is the average frequency of correct candidates among the generated candidates. Frequency Recall (**R**) is the average frequency of correct candidates among the gold candidates. Frequency **F1** is the harmonic average between P and R. Frequency metrics are shown in Eq. (6).

$$\mathrm{A_{Best}} = \frac{\sum_{best\ guess_i \in T} 1 \text{ if } best\ guess = m_i}{|T|} \tag{4}$$

$$\mathrm{A_{Oot}} = \frac{\sum_{a_i : i \in T} 1 \text{ if } any\ guess \in a_i = m_i}{|T|} \tag{5}$$

$$\mathrm{P} = \frac{\sum_{a_i : i \in T} \frac{\sum_{res \in c_i} freq_{res}}{A_i}}{|T|}, \mathrm{R} = \frac{\sum_{a_i : i \in T} \frac{\sum_{res \in c_i} freq_{res}}{H_i}}{|T|}, \mathrm{F1} = \frac{2 \times P \times R}{P + R} \tag{6}$$

4.3 Parameter tuning

For SCA-CLS substitute generation, we use the SemEval dataset as training data and set the parameter λ in the hybrid embedding strategy from 0 to 1 with an interval of 0.1. As shown in Fig. 3, SCA-CLS obtains the highest performance on A@1 and $\mathrm{A_{Best}}$ when λ equals 0.4. A@1 and $\mathrm{A_{Best}}$ evaluate the correctness of the first candidate, relating to user satisfaction, so we select 0.4 (A@1 = 0.410, $\mathrm{A_{Best}}$ = 0.356) as the optimal value of λ.

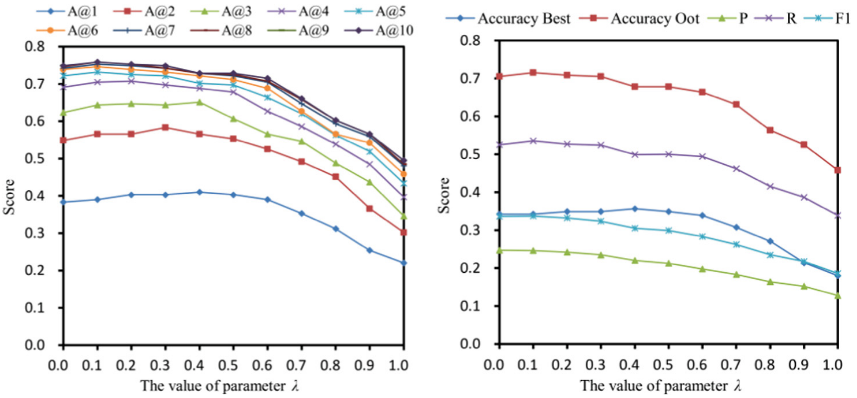


Fig. 3. Performance with different λ values from 0 to 1 with the interval as 0.1.

4.4 The Results

To evaluate the hybrid embedding strategy used in BERT for substitute generation, we study the effect of different input embedding strategies on NSP. As shown in Table 1, *Hybrid* is the hybrid embedding strategy we propose for the target word's embedding, which mixes up output embeddings returned from *5-gram* and *Gloss*. For the original sentence S and the sentence S' where the target word is replaced by "[MASK]", the input embedding strategies on NSP we use for comparison are shown as follows: *5-gram* is the strategy that we propose to optimize NSP in Sect. 3.3. *Gloss* is the strategy described in Sect. 3.1, that connects the gloss of target word and the sentence S' as BERT's input. *Mask* is the strategy using the sequence pair (S, S') connected by the sentence S and S' as input. 50%*Mask* is the strategy from LSBert, which randomly masked 50% of words in S, excluding the target word. *Keep* is the strategy using the sequence pair (S, S) connected by two sentences S as input.

Table 1. Comparison of different strategies for BERT to generate substitutes.

BERT Strategies	Dataset A						Dataset B					
	A@1	A@2	A@10	A_{Best}	A_{Oot}	F1	A@1	A@2	A@10	A_{Best}	A_{Oot}	F1
Hybrid	**0.410**	**0.550**	**0.735**	**0.265**	**0.602**	**0.380**	**0.437**	**0.563**	**0.790**	**0.429**	**0.782**	**0.644**
5-gram	0.390	0.498	0.719	0.245	0.594	0.371	0.378	0.496	0.782	0.370	0.765	0.590
Gloss	0.305	0.410	0.635	0.177	0.494	0.322	0.277	0.420	0.714	0.277	0.689	0.573
50% Masked	0.357	0.498	0.715	0.229	0.598	0.370	0.328	0.462	0.756	0.311	0.756	0.623
Mask	0.325	0.462	0.715	0.197	0.586	0.367	0.311	0.454	0.765	0.303	0.765	0.628
Keep	0.253	0.373	0.663	0.149	0.518	0.318	0.218	0.328	0.672	0.218	0.639	0.530

From the result, the *Keep* strategy returns the least satisfactory result, while the *5-gram* strategy is second to the *Hybrid*. This indicates that the model relies on the information carried by the input embedding to predict substitutes. However, when the information of the input embedding is too complete or repetitive, the model may be disturbed or overly dependent on that information. Our hybrid embedding strategy achieves the best performance in substitute generation, with A_{Best} increasing from 0.245 to 0.265 (8.2%) in Dataset A compared to the *5-gram* strategy, and with A_{Best} increasing from 0.277 to 0.429 (54.9%) in Dataset B compared to the *Gloss* strategy. It suggests that the hybrid strategy effectively incorporates both contextual and semantic information carried by the token embeddings of *5-gram* and *Gloss*.

We evaluate the performance of SCA-CLS on Dataset A and Dataset B. The results are shown in Tables 2 and 3. Our model outperforms the semantic-context ranking method [5] and context-aware method [23] on all A@N metrics on both Dataset A and B. Comparing to Context2vec and LSBert that also consider contextual information for LS, our model gains a better performance with A_{Best} improving by 208.1% (from 0.124 to 0.382) compared to Context2vec on Dataset A and A_{Best} improving by 101.0% (from 0.529 to 0.630) compared to LSBert on Dataset B. SCA-CLS achieves 30.8% and 57.0% higher than LexSubCon on A@1 on Dataset A and Dataset B, respectively.

While the state-of-the-art Merge-Sort model strengthens ranking by merging the BERT and Context2vec models, our model outperforms it in CLS, improving A_{Oot} from 0.538 to 0.667 (24.0%) and F1 from 0.373 to 0.415 (11.3%) on Dataset A, and A_{Oot} from 0.697 to 0.840 (20.5%) and F1 from 0.623 to 0.690 (10.8%) on Dataset B.

Table 2. Performance of all methods on the Dataset A and B using Accuracy@N.

Method	Dataset A									
	A@1	A@2	A@3	A@4	A@5	A@6	A@7	A@8	A@9	A@10
Semantic-context [5]	0.237	0.305	0.337	0.382	0.410	0.418	0.422	0.430	0.430	0.434
Context-aware [23]	0.261	0.325	0.365	0.394	0.410	0.418	0.438	0.438	0.446	0.450
Context2vec	0.181	0.233	0.257	0.257	0.265	0.273	0.273	0.273	0.273	0.273
LSBert	0.442	0.482	0.506	0.506	0.506	0.506	0.506	0.506	0.506	0.506
LexSubCon	0.390	0.558	0.622	0.647	0.675	0.691	0.703	0.715	0.719	0.723
Merge-Sort model	**0.522**	0.574	0.627	0.631	0.643	0.651	0.663	0.663	0.667	0.667
SCA-CLS	0.510	**0.582**	**0.647**	**0.691**	**0.699**	**0.719**	**0.735**	**0.751**	**0.755**	**0.759**
Method	Dataset B									
	A@1	A@2	A@3	A@4	A@5	A@6	A@7	A@8	A@9	A@10
Semantic-context [5]	0.261	0.311	0.361	0.403	0.429	0.437	0.437	0.437	0.437	0.437
Context-aware [23]	0.269	0.328	0.345	0.378	0.378	0.387	0.387	0.387	0.395	0.403
Context2vec	0.193	0.235	0.252	0.261	0.277	0.294	0.294	0.294	0.294	0.294
LSBert	0.538	0.613	0.630	0.639	0.655	0.655	0.655	0.655	0.655	0.655
LexSubCon	0.412	0.580	0.681	0.706	0.748	0.765	0.782	0.798	0.798	0.798
Merge-sort model	0.580	0.639	0.664	0.697	0.706	0.714	0.714	0.714	0.714	0.714
SCA-CLS	**0.647**	**0.731**	**0.782**	**0.790**	**0.798**	**0.807**	**0.807**	**0.815**	**0.832**	**0.840**

4.5 Ablation study

SCA-CLS combines four features to assist in ranking generated substitutes. To explore the effectiveness of each feature, we conduct an ablation study by removing one feature in turn. The results are shown in Table 4, where *Hybrid* represents the initial candidates (no ranking features are used) obtained by the hybrid embedding strategy. Arbitrarily removing one of the features obtains better results than *Hybrid*, and SCA-CLS that combines all four features has the best ranking performance, indicating that all features

Table 3. Performance comparison on the Dataset A and B using various evaluation metrics.

Method	Dataset A					Dataset B				
	A_{Best}	A_{Oot}	P	R	F1	A_{Best}	A_{Oot}	P	R	F1
Context2vec	0.124	0.189	0.163	0.142	0.152	0.193	0.286	0.257	0.270	0.263
LSBert	0.293	0.382	0.341	0.283	0.309	0.529	0.647	0.591	0.617	0.604
LexSubCon	0.257	0.627	0.325	0.487	0.390	0.395	0.790	0.569	0.788	0.661
Merge-Sort model	0.373	0.538	0.345	0.407	0.373	0.571	0.697	0.587	0.664	0.623
SCA-CLS	**0.382**	**0.667**	**0.353**	**0.502**	**0.415**	**0.630**	**0.840**	**0.604**	**0.805**	**0.690**

Table 4. Ablation study results of the ranking features.

Methods	Dataset A				Dataset B			
	A@1	A@2	A_{Best}	A_{Oot}	A@1	A@2	A_{Best}	A_{Oot}
Hybrid	0.410	0.550	0.265	0.602	0.437	0.563	0.429	0.782
SCA-CLS	**0.510**	**0.582**	**0.382**	**0.667**	**0.647**	**0.731**	**0.630**	**0.840**
w/o Gloss	0.506	0.578	0.369	0.659	0.639	0.731	0.613	0.840
w/o WordNet	0.486	0.566	0.353	0.639	0.622	0.723	0.597	0.824
w/o Complexity	0.506	0.582	0.357	0.639	0.563	0.697	0.546	0.815
w/o Frequency	0.494	0.570	0.345	0.647	0.546	0.672	0.538	0.832

positively affect ranking. As shown in the result, the impact of the gloss similarity feature on SCA-CLS is relatively small, which likely has the following two reasons: first, many substitutes fail to be recognized in WordNet so their glosses cannot be found; the second is that the gloss-selection model may introduce noise during selecting the best gloss for the substitute, resulting in obtaining inappropriate gloss.

5 Conclusions

This paper proposes a Semantic-Context-Aware framework SCA-CLS based on BERT for community-oriented lexical simplification. SCA-CLS develops a new hybrid embedding strategy to integrate gloss knowledge into contextual embeddings, which outperforms the previous state-of-the-art substitute generation strategy in LSBert. SCA-CLS introduces a gloss similarity feature to accurately calculate the semantic similarity between target words and substitutes by leveraging lexicon resources from WordNet, and a complexity feature that measures the complexity of the target word based on context. Compared to a list of baselines, experiment results show that SCA-CLS achieves the best performance in CLS tasks.

References

1. Feng, L.: Automatic readability assessment for people with intellectual disabilities. In: ACM SIGACCESS Accessibility and Computing, vol. 93, pp. 84–91 (2009)
2. Hirsh, D., Nation, P.: What vocabulary size is needed to read unsimplified texts for pleasure? Reading Foreign Lang. **8**(2), 689–696 (1992)
3. Nation, I.S.P.: Learning Vocabulary in Another Language. Cambridge University Press, Cambridge (2001)
4. De Belder, J., Moens, M.F.: Text simplification for children. In: SIGIR Workshop on Accessible Search Systems, pp. 19–26. ACM, New York (2010)
5. Hao, T., Xie, W., Lee, J.: A semantic-context ranking approach for community-oriented English lexical simplification. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 784–796. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_68
6. Education Bureau: Enhancing English Vocabulary Learning and Teaching at Secondary Level. http://www.edb.gov.hk/vocab_learning_sec. Accessed May 2020
7. Peters, M.E., Neumann, M., Iyyer, M., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Qiang, J., Li, Y., Zhu, Y., Yuan, Y., Wu, X.: LSBert: lexical simplification based on BERT. In: IEEE/ACM Transactions on Audio, Speech, and Language Processing, p. 99 (2021)
10. Li, R., Xie, W., Song, J., Wong, L.P., Wang, F.L., Hao, T.: A context-driven merge-sort model for community-oriented lexical simplification. In: 2022 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-ASIA), pp. 1–6 (2022)
11. Shardlow, M.: A survey of automated text simplification. Int. J. Adv. Comput. Sci. Appl. **4**(1), 58–70 (2014)
12. Devlin, S.: The use of a psycholinguistic database in the simplification of text for aphasic readers. Linguistic databases (1998)
13. Sinha, R.: UNT-SIMPRANK: systems for lexical simplification ranking. In: * SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pp. 493–496 (2012)
14. Nunes, B.P., Kawase, R., Siehndel, P., Casanova, M.A., Dietze, S.: As simple as it gets-a sentence simplifier for different learning levels and contexts. In: 2013 IEEE 13th International Conference on Advanced Learning Technologies, pp. 128–132. IEEE (2013)
15. Shardlow, M.: Out in the open: finding and categorising errors in the lexical simplification pipeline. In: LREC, pp. 1583–1590 (2014)
16. Glavaš, G., Štajner, S.: Simplifying lexical simplification: do we need simplified corpora? In: The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 63–68 (2015)
17. Melamud, O., Goldberger, J., Dagan, I.: context2vec: learning generic context embedding with bidirectional LSTM. In: the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 51–61 (2016)
18. Michalopoulos, G., McKillop, I., Wong, A., Chen, H.: LexSubCon: integrating knowledge from lexical resources into contextual embeddings for lexical substitution. In: The 60th Annual Meeting of the ACL, pp. 1226–1236 (2022)
19. Yap, B.P., Koh, A., Chng, E.S.: Adapting BERT for word sense disambiguation with gloss selection objective and example sentences. arXiv preprint arXiv:2009.11795 (2020)

20. Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a semantic concordance for sense identification. In: Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, 8–11 March (1994)
21. Ehara, Y., Miyao, Y., Oiwa, H., Sato, I., Nakagawa, H.: Formalizing word sampling for vocabulary prediction as graph-based active learning. In: EMNLP, pp. 1374–1384 (2014)
22. Lee, J.S., Yeung, C.Y.: Personalizing lexical simplification. In: The 27th International Conference on Computational Linguistics (COLING), pp. 224–232 (2018)
23. Song, J., Hu, J., Wong, L.-P., Lee, L.-K., Hao, T.: A new context-aware method based on hybrid ranking for community-oriented lexical simplification. In: Nah, Y., Kim, C., Kim, S.H., Moon, Y.-S., Whang, S.E. (eds.) DASFAA 2020. LNCS, vol. 12115, pp. 80–92. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59413-8_7
24. Song, J., Shen, Y., Lee, J., Hao, T.: A hybrid model for community-oriented lexical simplification. In: Zhu, X., Zhang, M., Hong, Y., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12430, pp. 132–144. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60450-9_11
25. Gooding, S., Kochmar, E.: Complex word identification as a sequence labelling task. In: The 57th Annual Meeting of the ACL, pp. 1148–1153 (2019)
26. Brysbaert, M., New, B.: Moving beyond Kuera and Francis: a critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. Behav. Res. Methods **41**, 977–990 (2009)
27. Paetzold, G.H., Specia, L.: A survey on lexical simplification. Int. J. Artif. Intell. Res. **60**, 549–593 (2017)
28. Sharoff, S.: Open-source corpora: using the net to fish for linguistic data. Int. J. Corpus Linguist. **11**(4), 435–462 (2006)
29. Horn, C., Manduca, C., Kauchak, D.: Learning a lexical simplifier using Wikipedia. In: The 52nd Annual Meeting of the ACL (Volume 2: Short Papers), pp. 458–463 (2014)

A Cross-lingual Sentiment Embedding Model with Semantic and Sentiment Joint Learning

Yuemei Xu[1](✉), Wanze Du[2], and Ling Hu[1]

[1] School of Information Science and Technology, Beijing Foreign Studies University, Beijing, China
xuyuemei@bfsu.edu.cn

[2] School of Software and Microelectronics, Peking University, Beijing, China

Abstract. Unsupervised cross-lingual word embedding (CLWE) aligns monolingual embedding spaces without parallel corpora or bilingual dictionaries. However, it is unclear whether CLWE models tuned for different corpora alignment can perform well across NLP tasks. Previous researches have shown that unsupervised CLWE tends to make words have close word embedding distributions with similar syntactic but opposite sentiment polarity, leading to poor performance in cross-lingual sentiment analysis (CLSA). This work proposes an Unsupervised Cross-lingual Sentiment word Embedding (UCSentiE) model, which eliminates both linguistic and sentiment gap between two languages. UCSentiE leverages the priori sentiment information of source language and integrates them into CLWE without compromising cross-lingual word semantics. We evaluate UCSentiE on two NLP tasks across six languages (English, Chinese, German, Japanese, French and Spanish). Experimental results demonstrate UCSentiE's stability in bilingual lexicon induction (BLI) and its superiority over unsupervised VecMap and supervised MUSE models in CLSA, with average F1 score improvements of about 6.53% and 2.23%, respectively. Visualization and clustering analysis further validates our approach's effectiveness. Code is available at https://github.com/dwzgit/UCSentiE.git.

1 Introduction

Cross-lingual word embedding (CLWE) plays a crucial role in facilitating cross-lingual knowledge transfer between resource-rich and low-resource languages [1]. CLWE aligns monolingual embeddings via bilingual dictionaries or parallel corpora, causing similar words in diverse languages to draw close in the shared CLWE space [2].

Recently, unsupervised CLWE approaches [3–5] have been proposed, which do not rely on bilingual dictionaries or parallel corpora, but instead use techniques such as Generative Adversarial Networks (GANs) [6], Optimal Transport

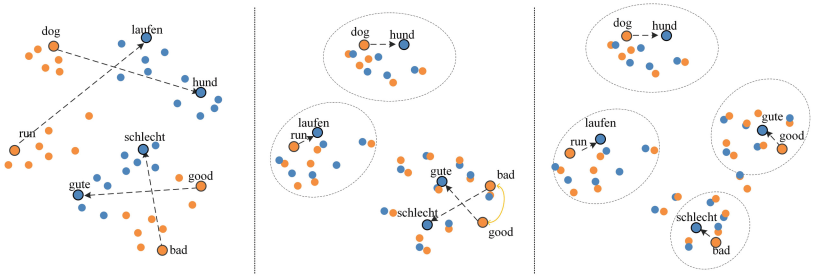
This work was supported by the Fundamental Research Funds for the Central Universities (No. 2022JJ006).

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 82–94, 2023.
https://doi.org/10.1007/978-3-031-44693-1_7

Theory (OPT) [7], and the iterative closest point (ICP) algorithm [8] to align monolingual embedding spaces. Unsupervised CLWE reduces reliance on bilingual dictionaries or parallel corpora and performs well on close language pairs, such as English-German. However, there are still two critical issues that merit further investigation.

Firstly, whether the cross-lingual word embeddings tuned for different corpora alignments can achieve equally good performance on different NLP tasks. For example, fastText, a cross-lingual word embedding model trained on Wikipedia multilingual datasets, offers 157 language word embedding representations. But fastText performs worse than monolingual word embedding models like Word2Vec in some NLP tasks (i.e., named entity recognition).

Secondly, whether incorporating more information into CLWE can enhance NLP task performance while maintaining word semantics and task-specific features. Existing studies indicate that incorporating sentiment information into monolingual word embeddings can significantly boost sentiment classification performance [9–11]. However, it is more challenging for unsupervised CLWE to achieve a balance between cross-lingual semantic alignment and sentiment integration, especially when only the source language has annotated sentiment data.



(a) Monolingual word embeddings (b) Cross-lingual semantic alignment (c)Cross-lingual semantic and sentiment alignment

Fig. 1. Comparison of CLWE spaces using different alignment methods

Figure 1 provides an illustration of English-German CLWE spaces, with English words marked in yellow and German in blue. Figure 1(a) shows the monolingual word embedding space, where words with similar semantics are not aligned. Figure 1(b) shows the CLWE space after semantic alignment, where words such as "run", "dog" and their corresponding German translations "laufen" and "hund" are now aligned. However, the alignment of sentiment words like "good", "bad", "gute", "schlecht" is suboptimal. Figure 1(c) displays the CLWE space after joint learning of semantics and sentiment, which is advantageous in aligning sentiment words without sacrificing their semantic alignment.

Based on the aforementioned discussion, this paper proposes an unsupervised cross-lingual word embedding model with jointly learning word semantic and sentiment information, called UCSentiE. The proposed model incorporates sentiment information from annotated data in the source language during the initial projection iteration without compromising cross-lingual word semantic. Upon the initial projection process, the embeddings of corresponding words in the source and target languages become highly similar. Thus, the integration of sentiment information into the source language word embeddings parallels its incorporation into bilingual word embeddings, facilitating the transfer of sentiment knowledge from the source language to the target language.

We conducted experiments on 6 languages and compared UCSentiE with the existing models [4,12–14] in bilingual lexicon induction (BLI) and cross-lingual sentiment analysis (CLSA) tasks. Experimental results demonstrate that UCSentiE can achieve cross-lingual word semantic alignment while preserving sentiment information. The main contributions of the paper are as follows:

– We propose an unsupervised CLWE model to jointly learn word semantic and sentiment information. Prior sentiment information from the source language is transferred to CLWE representation without sacrificing the alignment of word semantic.
– We carry out extensive experiments and visualization analysis, and find that CLWE models are sensitive to language pairs, with better performance on close language pairs, such as English-Spanish, compared to distant language pairs, such as English-Japanese.
– We analyze the factors that affect cross-lingual alignment and find that language distance and monolingual word embedding quality are important. Sole reliance on the unsupervised CLWE model is not enough to achieve high-quality alignment for distant language pairs. Our comparison of fastText and Word2Vec models within CLWE reveals fastText's superiority. Furthermore, the quality of monolingual word embeddings also affects CLWE performance. For instance, the English-Spanish pair outperforms the seemingly semantically closer English-German pair, possibly due to the low-quality German monolingual word embeddings that negatively impacted the model's performance.

2 Related Work

2.1 Cross-lingual Word Embedding

Cross-lingual word embedding models can be classified into three research categories based on their use of annotated data: supervised, semi-supervised, and unsupervised methods.

Early CLWE models were primarily based on supervised methods that heavily relied on human-annotated data to align the monolingual word embedding of the source and target languages. However, obtaining such annotated data can be challenging for most languages. As a result, semi-supervised CLWE approaches were developed to mitigate the reliance on cross-lingual supervision. These methods utilize smaller amounts of annotated data or seed dictionaries, i.e., 25 translated word pairs as heuristic seed dictionary [15].

Introduced in 2015 [16], Unsupervised CLWE approaches have now become mainstream in the field of CLWE [4,13,17–19] as they do not require any parallel corpora or seed dictionaries. By leveraging large-scale non-parallel corpora, unsupervised methods employ models like Generation Adversarial Networks, Auto-Encoder-Decoder and others, to mine language relationships.

Conneau et al. proposed an unsupervised adversarial model called MUSE [4], which has achieved promising results and even outperformed supervised approaches in some cases. Mohiuddin et al. revisited adversarial models and add two regularization terms that yield improved results [19]. Artetxe et al. proposed an unsupervised self-learning model called VecMap based on a semi-supervised approach [13], which achieved 48% accuracy in English-German pairs, and 37% accuracy in English-Spanish pair. Li et al. proposed Iterative Dimension Reduction to improve the robustness of VecMap [18]. Shen et al. used Adversarial Auto-Encoder (AAE) to learn bilingual parallel texts and then used a BiGRU model to achieve CLWE alignment via a linear transformation matrix [17]. Rasooli et al. considered the influence of language families on CLWE, and introduced a multi-source approach to reduce the gap between languages [20].

Unsupervised CLWE approaches often assume isometric monolingual word embedding space across different languages [21]. However, this isometry assumption does not hold true for all language pairs, particularly when dealing with semantically distant languages [22]. Additionally, the existing unsupervised CLWE approaches only consider semantic information, and it remains to be investigated whether incorporating additional information from various NLP tasks can still maintain good performance on CLWE alignment.

2.2 Sentimental Embedding

Sentimental embeddings learn word representations that encode both syntactic context and sentiment polarity. Maas et al. first proposed to leverage document-level sentiment annotated data to obtain word-level sentiment embedding [23]. Later, Tang et al. proposed a sentiment-specific word embedding model to generate word embeddings that incorporate sentiment information [24].

Inspired by Maas and Tang's work, researches extended sentiment embedding from single-language to multiple languages and investigated cross-lingual sentiment word embedding. Abdalla et al. found that sentiment information is still highly preserved after aligning the two monolingual embedding spaces, even with low-quality seed bilingual dictionaries [25]. Building upon the research in [25], Dong proposed a CLWE model to improve adaption in cross-lingual sentiment classification tasks by incorporating sentiment information [26]. By leveraging annotated bilingual parallel corpora, he encoded the latent sentiment information into vectors, producing sentiment embeddings for many languages. These sentiment embeddings were then integrated with word embeddings using a Dual-channel Convolutional Neural Network (DC-CNN). Barnes presented an approach that incorporates sentiment information from a bilingual sentiment dictionary into the shared vector space. This integration enables the retention of sentiment features from both languages [27]. On the other hand, Ma et al.

proposed an Unsupervised Bilingual Sentiment word Embedding (UBSE) using Generative Adversarial Nets, which relies solely on a sentiment lexicon of the source language [28].

However, these methods utilized machine translation tools [25], bilingual parallel corpora [26], or bilingual dictionaries [27] to incorporate sentiment information within a single language, which were not suitable for cross-lingual word embedding alignment, particularly for resource-scarce languages lacking sentiment-annotated data.

3 Proposed Method

As illustrated in Fig. 2, the overall structure of UCSentiE consists of three main modules to learn the mapping matrix $\mathbf{W}$ from the source language to the target language: initial solution generation (3.1), priori sentimental information embedding represented by the sentiment mapping matrix $\mathbf{W}_{senti}$ (3.2), and sentiment-aware CLWE self-learning iterations that update the semantics mapping matrix $\mathbf{W}_{seman}$ by integrating $\mathbf{W}_{senti}$ (3.3).

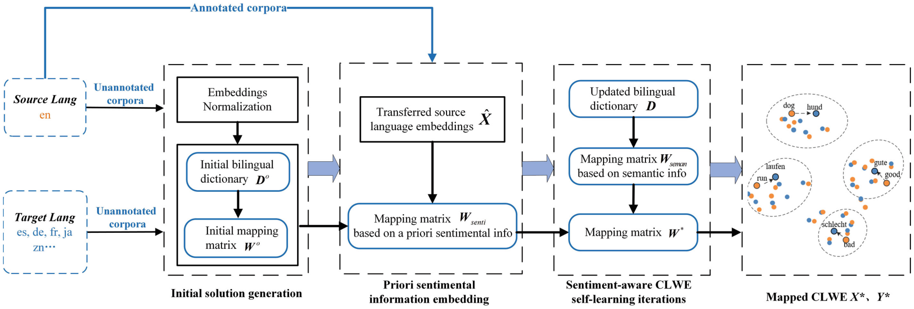

Annotated corpora
Source Lang
en
Unannotated corpora
Target Lang
es, de, fr, ja
zn···
Unannotated corpora
Embeddings Normalization
Initial bilingual dictionary D^o
Initial mapping matrix W^o
Initial solution generation
Transferred source language embeddings $\hat{X}$
Mapping matrix W_{senti} based on a priori sentimental info
Priori sentimental information embedding
Updated bilingual dictionary D
Mapping matrix W_{seman} based on semantic info
Mapping matrix W^*
Sentiment-aware CLWE self-learning iterations
dog
hund
laufen
run
gute
good
schlecht
bad
Mapped CLWE X^*、Y^*

Fig. 2. A flowchart of UCSentiE model

3.1 Initial Solution Generation

The normalized monolingual embeddings in the source and target languages are denoted as $\overline{\mathbf{X}} \in \mathbb{R}^{m \times d}$ and $\overline{\mathbf{Y}} \in \mathbb{R}^{n \times d}$, where the i-th row of $\overline{\mathbf{X}}(\overline{\mathbf{Y}})$ corresponds to the vector representation of the i-th word. Here, m and n are the vocabulary sizes, respectively. The normalization process contains length normalization, mean centering, and length scaling [29].

The normalized embeddings are not aligned along the first axis, i.e., the i-th row of $\overline{\mathbf{X}}$ does not correspond to the i-th row of $\overline{\mathbf{Y}}$. Following [13], we establish an initial bilingual dictionary as the basis. We first compute the similarity matrices of $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ and sort the values in each row to obtain $\mathbf{sorted}(\mathbf{s_x})$ and $\mathbf{sorted}(\mathbf{s_y})$. Given a word v_i^s and its row in $\mathbf{sorted}(\mathbf{s_x})$, we apply CSLS retrieval over the rows of $\mathbf{sorted}(\mathbf{s_y})$ to find its corresponding translation v_j^t, yielding an initial mapping dictionary $D^0 = \{(v_1^s, v_1^t), (v_2^s, v_2^t), ..., (v_K^s, v_K^t)\}$, where v_j^t is the translation of the source word v_i^s and K is the number of the initial dictionary. Given an initial bilingual seed dictionary D^0, we can learn the initial projection matrix $\mathbf{W}^0$ by forcing word pairs in D^0 to have similar representations in the bilingual space.

3.2 Priori Sentimental Information Embedding

In order to incorporate the priori sentiment information of the source language into the CLWE, we calculate an priori sentimental mapping matrix $\mathbf{W}_{senti}$ to enable sentiment transfer between two languages. The sentiment annotated corpora in the source language is denoted as $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\}$, with $\mathbf{s}_i$ representing the i-th document and N being the total number of documents in $\mathbf{s}$. The sentiment labeling is denoted by $\mathbf{l} = \{l_1, l_2, ..., l_N\}$ with $l_i = 0$ indicating a positive sentiment and $l_i = 1$ indicating a negative sentiment for the i-th document.

Given the initial bilingual seed dictionary D^0, we derive the mapped source word embedding in the bilingual space by calculating $\hat{\mathbf{x}}_i = \mathbf{W}_{senti}\overline{\mathbf{x}_i}$, where $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_K\}$ and $\mathbf{W}_{senti}$ starts as $\mathbf{W}^0$. Our goal is to imbue the mapped source language word embeddings $\hat{\mathbf{X}}$ with sentiment information such that the sentiment annotated corpora represented by $\hat{\mathbf{X}}$ can be effectively utilized for predicting sentiment polarity. Let $\hat{l}_i$ denote the sentiment classifier used for predicting the sentiment polarity of $\mathbf{s}_i$. Thus, the sentiment prediction loss function can be expressed as:

$$\mathcal{L}_{pred} = -\sum_{i=1}^{N} l_i \log \hat{l}_i - (1 - l_i)\log(1 - \hat{l}_i) \tag{1}$$

To ensure that the translation pairs in the bilingual seed dictionary D^0 remain as similar as possible even after sentiment information incorporation, we introduce a mapping loss function calculated as:

$$\mathcal{L}_{proj} = \frac{1}{K}\sum_{i=1}^{K}(\hat{\mathbf{x}}_i - \overline{\mathbf{y}}_i)^2 \tag{2}$$

The overall loss function is as follow:

$$\mathcal{L} = \alpha \times \mathcal{L}_{pred} + (1 - \alpha) \times \mathcal{L}_{proj} \tag{3}$$

where a hyper-parameter α is used to balance the relative importance of sentiment and semantics, ensuring that the incorporation of sentiment information as a mapping guide does not compromise the mapping quality. By minimizing the total loss function within several training iterations, we obtain the priori sentiment mapping matrix $\mathbf{W}_{senti}$.

3.3 Sentiment-Aware CLWE Self-learning Iterations

Unsupervised sentiment-aware CLWE generation is a self-learning iterative procedure that considers both semantic and sentiment information of the language in each iteration represented by $\mathbf{W}_{senti}$ and $\mathbf{W}_{seman}$, respectively. Our method involves updating D, $\mathbf{W}_{seman}$ and $\mathbf{W}$ through a self-learning procedure in the following three steps:

1. Update $\mathbf{W}_{seman}$: In each iteration, we first generate two sorted matrices $\mathbf{X}_D$ and $\mathbf{Y}_D$, based on the current dictionary D. The i-th row of $\{\mathbf{X}_D, \mathbf{Y}_D\}$ corresponds to the i-th word pair in D. The optimal $\mathbf{W}_{seman}$ is obtained through singular value decomposition (SVD) as follows:

$$\begin{aligned} \mathbf{W}_{seman} &= \mathbf{V} \cdot \mathbf{U}^{\mathsf{T}} \\ \mathbf{U} \textstyle\sum \mathbf{V}^{\mathsf{T}} &= SVD(\mathbf{Y}_D^{\mathsf{T}} \cdot \mathbf{X}_D) \end{aligned} \tag{4}$$

2. Update mapping matrix $\mathbf{W}$: Given a hyper-parameter β, we calculate the cross-lingual mapping matrix $\mathbf{W}$ in each iteration by combining $\mathbf{W}_{senti}$ and $\mathbf{W}_{seman}$:

$$\mathbf{W} = \beta \times \mathbf{W}_{senti} + (1 - \beta) \times \mathbf{W}_{seman} \tag{5}$$

   where the trade-off between the two matrices is controlled by $\beta \in [0, 1]$. When $\beta = 0$, no priori sentiment information is carried in $\mathbf{W}$; When $\beta = 1$, no semantic information is carried in $\mathbf{W}$. We solve the optimal mapping matrix $\mathbf{W}$ by maximizing the similarities for the current dictionary D:

$$\arg\max_{W} \left(\sum_i \sum_j D_{ij} (\overline{\mathbf{x}}_i \mathbf{W}) \cdot \overline{\mathbf{y}}_j\right) \tag{6}$$

3. Update bilingual dictionary D: Given a computed $\mathbf{W}$, we can generate a new dictionary D by using CSLS retrieval over the rows of $\overline{\mathbf{X}}\mathbf{W}$ and $\overline{\mathbf{Y}}$. The updated bilingual dictionary D can be used to create new sorted matrices $\mathbf{X}_D$ and $\mathbf{Y}_D$, and the iterative process continues.

The iterative training process stops when the objective function (6) is optimally solved, resulting in the optimal $\mathbf{W}^*$, which allows us to obtain the final sentiment-aware CLWE and a bilingual dictionary D.

4 Experiments

4.1 Evaluation Tasks and Baselines

Experiments evaluate the performance of UCSentiE on two tasks: Bilingual Lexicon Induction (BLI) and Cross-lingual Sentiment Analysis (CLSA). BLI involves constructing bilingual dictionaries using the generated cross-lingual word embeddings and comparing them with existing bilingual dictionaries, while CLSA involves predicting sentiment in target language texts using the generated embeddings.

The experiments cover language pairs from English(En) to Spanish(Es), German(De), French(Fr), Japanese(Ja), and Chinese(Zn), and compare UCSentiE against four models: (1) Monolingual approach: a multilingual word embedding model with 157 languages, fastText, that is applied directly to the tasks [14]. For brevity, we refer to this approach as "Mono". (2) MUSE: a supervised model that learns an orthogonal map from the source to the target embedding space via supervised translation pairs [4]. (3) VecMap: an unsupervised model that assumes the monolingual space is isometric across languages and focuses solely on semantics [13]. (4) XU: an unsupervised bidirectional transformation model using Sinkhorn distance, with transformation direction from other languages to English [12].

4.2 Datasets and Evaluation Metrics

For the BLI task, we use the MUSE bilingual lexicon released by Facebook[1] as the baseline lexicon. The MUSE lexicon is a collection of 110 bilingual lexicons, including pairs such as English-Spanish. Each lexicon contains the 6500 most frequently used words in each language. For the CLSA task, we utilize the Amazon product review dataset[2] for English, German, French, Japanese, and Chinese, consisting of $6,000$ positive and $6,000$ negative reviews for each language. We use the Spanish review dataset in [27], which comprises 1,472 user reviews (1,216 positive and 256 negative), as the Amazon dataset lacks Spanish reviews. The data was divided into training and testing sets at a 3:2 ratio for each language in the experiments.

In the BLI task, we compare each model's bilingual lexicon to the benchmark MUSE lexicon and evaluate their performance using the accuracy rate (Acc) and precision P@N metrics (for $N = 1, 5$). The P@N metric measures the accuracy of the N-th nearest neighbor's translation for a given source word, using the CSLS method. In the CLSA task, we use cross-lingual word embeddings to perform multilingual sentiment classification and evaluate the performance using metrics of accuracy (Acc) and F1-score.

We use 300-dimensional fastText embeddings[3] as monolingual word embeddings by default and compare the impact of monolingual word embeddings on the model in later subsections.

4.3 Experiment Results

Results in BLI Task. Table 1 presents the accuracy comparison of the induced bilingual dictionaries across different methods. The best accuracy score is highlighted in **bold**, the suboptimal score is highlighted in underlined. UCSentiE outperforms all the baseline methods on three of five language pairs and

[1] https://github.com/facebookresearch/MUSE.

[2] http://jmcauley.ucsd.edu/data/amazon/.

[3] https://github.com/facebookresearch/fastText.

Table 1. Accuracy comparison of BLI in different models.

Model	N = 5					N = 1				
	En-Es	En-De	En-Fr	En-Ja	En-Zn	En-Es	En-De	En-Fr	En-Ja	En-Zn
Mono [14]	68.06	59.16	68.69	0.01	0.00	56.23	44.21	54.68	0.00	0.00
XU [12]	67.54	59.56	68.18	0.01	0.07	55.85	45.01	54.04	0.00	0.02
MUSE [4]	68.18	59.44	67.83	28.66	**54.66**	56.36	44.48	53.65	18.99	**31.49**
VecMap [13]	68.74	60.45	**73.64**	50.87	48.63	**57.19**	**45.93**	**56.73**	29.75	28.21
UCSentiE	**68.79**	**60.81**	68.71	**52.27**	54.31	56.65	45.68	54.30	**31.98**	27.15

achieves suboptimal scores in the remaining pairs when $N = 5$. It verifies that the sentiment-aware cross-lingual word embeddings generated by the UCSentiE model maintain semantic information and still perform outstandingly in the BLI Task. Observing the performance drop as P@N decreases from $N = 5$ to $N = 1$ across all models. However, UCSentiE still achieves optimal performance on En-Ja, and sub-optimal performance on En-Es and En-De, which indicates its capacity to preserve comprehensive word semantics while incorporating sentiment. Note that the unsupervised VecMap model performs better than the supervised MUSE model due to the absence of any supervision information provided to MUSE.

We observe the performance difference across language pairs. In all methods, BLI performances on En-Ja and En-Zn are lower than those on the rest, perhaps because English is relatively distant to Japanese and Chinese. Moreover, the monolingual approach's performance on En-Ja and En-Zn is nearly zero, indicating the inability of monolingual vectors to achieve alignment on distant language pairs and cross-lingual embeddings are the basis of knowledge transfer.

Another observation is that the monolingual vector quality significantly influences model performance. Among the language pairs, the highest accuracy of BLI should be achieved on En-De out of the most similar grammar and pronunciation rather than En-Es and En-Fr on all the methods. The unexpected result indicates that the monolingual word vectors of English and German are poorly aligned, leading to a degraded performance of cross-lingual alignment.

Results in CLSA Task. Table 2 summarizes the results of all the methods in the CLSA task. UCSentiE outperforms all the methods on En-En, indicating that English word vectors aligned by UCSentiE across languages can still perform well in monolingual sentiment classification tasks. On cross-lingual experiments, UCSentiE contains at least one optimal or suboptimal F1 score or accuracy rate on each language pair, verifying the good performance of UCSentiE in the CLSA task after embedding sentiment information.

When comparing model performance across different language pairs, we obtained similar conclusions within the BLI task. The models exhibit higher performance on language pairs with closer linguistic distance, such as En-Es and En-Fr, in contrast to pairs like En-Zn and En-Ja, which have greater linguistic distance. This indicated that the semantic distance between languages also has an impact on CLSA. In addition, all the models outperform on En-Es while their

Table 2. Performance comparison of CLSA in different models.

Model	F1						Acc					
	En-En	En-Es	En-De	En-Fr	En-Ja	En-Zn	En-En	En-Es	En-De	En-Fr	En-Ja	En-Zn
Mono [14]	74.04	88.54	60.24	71.99	0.10	66.66	74.88	79.44	68.85	69.03	50.00	49.99
XU [12]	75.15	**88.71**	**66.67**	66.67	66.68	20.23	75.60	**79.91**	50.00	50.00	50.02	49.78
MUSE [4]	74.75	88.54	60.24	66.68	66.64	66.66	75.38	79.44	68.85	50.02	49.98	49.99
VecMap [13]	74.34	85.80	62.63	72.73	**66.68**	69.05	74.82	77.57	69.63	**72.05**	50.10	57.67
UCSentiE	**75.97**	87.28	66.16	**74.36**	64.73	**71.76**	**76.50**	79.44	**71.23**	71.95	**60.00**	**74.93**

performance on En-De is less satisfactory, and the F1 scores of En-De are even slightly lower than those on En-Zn, which may be due to the influence of the quality of monolingual word vectors in German on the model performance.

Impact of Monolingual Word Embeddings. Previous studies highlighted the impact of monolingual word embeddings on CLWE performance [4]. To assess the effects of various monolingual embeddings on our model's performance, we contrast results using 300-dimensional fastText embeddings and Word2vec embeddings, and refer to the two variants of our model as UCSentiE(F) and UCSentiE(W), respectively.

Table 3. Impact of Monolingual Word Embeddings.

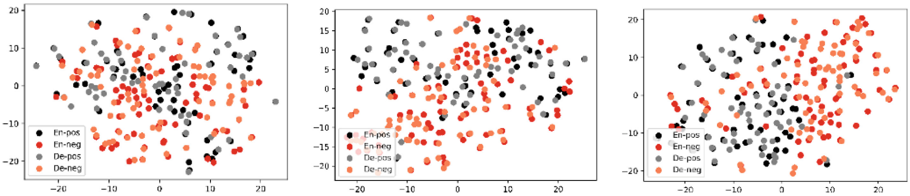
	En-En	En-Es	En-De	En-Fr	En-Ja	En-Zn	En-En	En-Es	En-De	En-Fr	En-Ja	En-Zn
BLI	N = 5						N = 1					
UCSentiE(F)	-	**68.79**	60.81	**68.71**	**52.27**	**54.31**	-	**56.65**	45.68	**54.30**	31.98	**27.15**
UCSentiE(W)	-	52.97	**65.02**	55.52	47.31	25.02	-	38.12	**49.22**	39.16	**33.27**	15.45
CLSA	F1						Acc					
UCSentiE(F)	75.97	**87.28**	66.16	**74.36**	64.73	**71.76**	76.50	**79.44**	71.23	**71.95**	60.00	**74.93**
UCSentiE(W)	**77.04**	75.26	**73.21**	53.46	**66.97**	52.73	**77.40**	66.82	**74.55**	50.02	**67.30**	66.40

Results in Table 3 demonstrated that, in the BLI task, the accuracy of UCSentiE(F) is higher than that of UCSentiE(W) on four out of five language pairs at $N = 5$ and on three pairs at $N = 1$. This verifies that the choice of monolingual word vectors impacts cross-lingual word embedding alignment. Generally, CLWE derived from fastText perform better than that of Word2vec, but the specific performance of both varies with the target language. In the CLSA tasks, the average F1 value of UCSentiE(F) on all language pairs is 6.93% higher than that of UCSentiE(W), indicating that CLWE derived from fastText have an overall better performance than Word2vec. This aligns with findings from the BLI tasks.

4.4 Visualization Analysis

To illustrate the effectiveness of the UCSentiE model in balancing both semantics and sentiment, Fig. 3 uses English-German language pair as an example to perform visualization analysis. We employ principal component analysis (PCA) to map the 300-dimensional word embeddings to a 2-dimensional plane. We randomly selected 140 positive and negative sentiment words from SentiWordnet[4] and visualized them in CLWE spaces of fastText, VecMap, and UCSentiE, respectively. In each CLWE space, every word is represented as a dot, where positive(negative) source language words are marked in black(red), positive(negative) target language words are marked in gray(orange).

Figure 3(a) shows the CLWE space in fastText, where some English words and their corresponding German translations are relatively close (the overlapping of red-orange dots and black-gray dots). This indicates that the English-German word embeddings in fastText are partially aligned. Comparatively, the Vecmap and UCSentiE models in Figs. 3(b) and 3(c) bring the word embeddings of English and German even closer in the vector space. Upon observing Fig. 3(c), we can find that most black points correspond to gray points, and red points correspond to orange points. This finding shows that words sharing the same sentiment polarity are closely clustered in the UCSentiE space.



(a) CLWE in fastText. (b) CLWE in VecMap. (c) CLWE in UCSentiE.

Fig. 3. Visualization of English-German word embeddings in different CLWE spaces. En-pos(En-neg) denotes positive(negative) words in English. De-pos(De-neg) denotes positive(negative) words in German.

5 Conclusion

We propose an unsupervised cross-lingual sentiment word embedding model that utilizes priori sentiment information from the source language to consider both semantic and sentiment information. We empirically evaluate our method on tasks of bilingual lexicon induction and cross-lingual sentiment analysis on five target languages using English as the source language. Also, the experiments compare the impact of monolingual word embeddings on CLWE model. Visualization and cluster analysis illustrates the effectiveness of our proposed model.

[4] https://github.com/aesuli/SentiWordNet.

References

1. Cao, S., Kitaev, N., Klein, D.: Multilingual alignment of contextual word representations. In: ICLR (2020)
2. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013)
3. Artetxe, M., Labaka, M., Agirre, E.: Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In: AAAI, pp. 5012–5019 (2018)
4. Conneau, A., Lample, G., Ranzato, M.A., et al.: Word translation without parallel data. In: ICLR (2018)
5. Conneau, A., Khandelwal, K., Goyal, N., et al.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial networks. In: NIPS, pp. 2672–2680 (2014)
7. Alvarez-Melis, D., Jaakkola, T.S.: Gromov-wasserstein alignment of word embedding spaces. In: EMNLP, pp. 1881–1890 (2018)
8. Hoshen, Y., Wolf, L.: Non-adversarial unsupervised word translation. In: EMNLP, pp. 469–478 (2018)
9. Ke, P., Ji, H., Liu, S., et al.: Sentilare: sentiment-aware language representation learning with linguistic knowledge. In: EMNLP, pp. 6975–6988 (2020)
10. Wang, Y., Huang, G., Li, J., et al.: Refined global word embeddings based on sentiment concept for sentiment analysis. IEEE Access **9**, 37075–37085 (2021)
11. Liu, J., Zheng, S., Xu, G., et al.: Cross-domain sentiment aware word embeddings for review sentiment analysis. Int. J. Mach. Learn. Cybern. **12**, 343–354 (2020)
12. Xu, R., Yang, Y., Otani, N., et al.: Unsupervised cross-lingual transfer of word embedding spaces. In: EMNLP, pp. 2465–2474 (2018)
13. Artetxe, M., Labaka, G., Agirre, E.: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In: ACL, pp. 789–798 (2018)
14. Armand, J., Edouard, G., Piotr, B., et al.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
15. Artetxe, M., Labaka, G., Agirre, E.: Learning bilingual word embeddings with (almost) no bilingual data. In: ACL, pp. 451–462 (2017)
16. Gouws, S., Bengio, Y., Corrado, G.: Bilbowa: fast bilingual distributed representations without word alignments. In: ICML, pp. 748–756 (2015)
17. Shen, J., Liao, X., Lei, S.: Cross-lingual sentiment analysis via aae and bigru. In: IPEC, pp. 237–241 (2020)
18. Li, Y., Luo, Y., Lin, Y., et al.: A simple and effective approach to robust unsupervised bilingual dictionary induction. In: COLING, pp. 5990–6001 (2020)
19. Mohiuddin, T., Joty, S.R.: Revisiting adversarial autoencoder for unsupervised word translation with cycle consistency and improved training. In: NAACL, pp. 3857–3867 (2019)
20. Rasooli, M.S., Farra, N., Radeva, A., et al.: Cross-lingual sentiment transfer with limited resources. Machine Translation, pp. 143–165 (2018)
21. Vulić, I., Ruder, S., Søgaard, A.: Are all good word vector spaces isomorphic? In: EMNLP, pp. 3178–3192 (2020)
22. Glavaš, G., Vulić, I.: Non-linear instance-based cross-lingual mapping for non-isomorphic embedding spaces. In: ACL, pp. 7548–7555 (2020)
23. Maas, A.L., Daly, R.E., Pham, P.T., et al.: Learning word vectors for sentiment analysis. In: ACL, pp. 142–150 (2011)

24. Tang, D., Wei, F., Yang, N., et al.: Learning sentiment-specific word embedding for twitter sentiment classification. In: ACL, pp. 1555–1565 (2014)
25. Abdalla, M., Hirst, G.: Cross-lingual sentiment analysis without (good) translation. In: IJCNLP, pp. 506–515 (2017)
26. Dong, X., De Melo, G.: Cross-lingual propagation for deep sentiment analysis. In: AAAI, pp. 5771–5778 (2018)
27. Barnes, J., Klinger, R., Walde, S.S.I.: Bilingual sentiment embeddings: joint projection of sentiment across languages. In: ACL, pp. 2483–2493 (2018)
28. Ma, C., Xu, W.: Unsupervised bilingual sentiment word embeddings for cross-lingual sentiment classification. In: ICIAI, pp. 180–183 (2020)
29. Artetxe, M., Labaka, G., Agirre, E.: Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In: EMNLP, pp. 2289–2294 (2016)

Learning to Attentively Represent Distinctive Information for Semantic Text Matching

Junjie Wang, Rui Peng, and Yu Hong(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
tianxianer@gmail.com

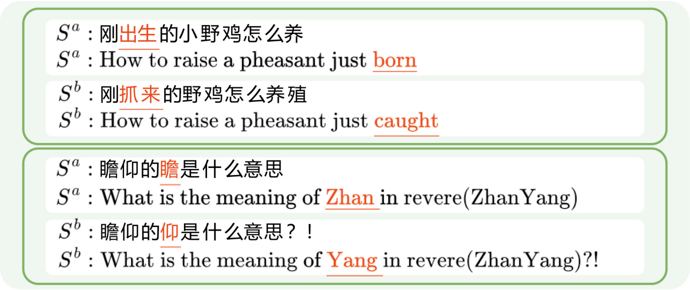
Abstract. Pre-trained language models (PLMs) such as BERT have achieved remarkable results in the task of Semantic Text Matching (STM). Nevertheless, existing models face challenges in discerning subtle distinction between texts, although it is the vital clue for STM. Concretely, the alteration of a single word causes significant variation of semantics of the entire text. To solve the problem, we propose a novel method of attentively representing distinctive information for STM. It comprises two components, including Reversed Attention Mechanism (RAM) and Sample-based Adaptive Learning (SAL). RAM reverses the hidden states of texts before computing attentions, which contributes to the highlighting of mutually-different syntactic constituents when comparing texts. In addition, during the initial stage of training, the model may acquire some biases. For example, it may ignore the distinctions between sentence pairs and simply classify sentence pairs with high lexical overlap as positive examples, because a majority of positive examples exhibit such high lexical overlap. SAL is designed to facilitate the model in comprehensively acquiring the semantic knowledge hidden in the distinctive constituents. Experiments on six STM datasets demonstrate the effectiveness of our proposed approach. Furthermore, we employ ChatGPT to generate textual descriptions of distinction between texts and empirically validate the significance of distinctive information in semantic text matching task.

Keywords: Text semantic matching · Pre-trained language model · Paraphrase identification

1 Introduction

Semantic Text Matching (STM) is a task of automatically determining whether two sentence-level texts are semantically similar. It is useful for information retrieval and factoid question answering. Recently, significant advancements have been made in this field due to the use of pre-trained language models (PLMs), such as BERT [6] and RoBERTa [10]. The major advantage of PLMs-based STM models is the ability of representing the deep semantics of sentences, which mostly benefits from the pretraining over large data and cyclopaedic knowledge. However, the previous studies still suffer from the difficulty in perceiving and representing subtle differences hidden in the sentences. Figure 1 shows a couple of examples, which

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 95–107, 2023.
https://doi.org/10.1007/978-3-031-44693-1_8



S^a : 刚出生的小野鸡怎么养
S^a : How to raise a pheasant just born

S^b : 刚抓来的野鸡怎么养殖
S^b : How to raise a pheasant just caught

S^a : 瞻仰的瞻是什么意思
S^a : What is the meaning of Zhan in revere(ZhanYang)

S^b : 瞻仰的仰是什么意思？！
S^b : What is the meaning of Yang in revere(ZhanYang)?!

Fig. 1. Sentence pairs with high lexical overlap but different semantics.

are semantically-dissimilar to each other although possessing inconspicuous difference. The fundamental bottleneck is that the PLMs-based STM models heavily rely on lexical consistency from the perspective of pragmatics. Specifically, Li et al. [8] establish a theoretical connection between BERT's pre-training objective and its non-smooth anisotropic semantic embedding space. This results in a strong dependence on lexical similarity when determining semantic similarity [21,23,25]. This phenomenon changes to be more obvious when sentences contain a large number of overlapped syntactic constituents. (Fig. 1)

In this paper, we attempt to enhance the BERT-based STM model by improving its ability of highlighting distinctive syntactic constituents. Specifically, we propose Reversed Attention Mechanism (RAM) and Sample-based Adaptive Learning (SAL), and use them to optimize the fine-tuning process when transferring BERT to task-specific data. RAM is designed to relatively effortless perception on subtle distinction between sentences, and provide high attentions upon the inconsistent words when computing their hidden states. SAL is used to reduce the bias that distracts the training process, where the serious bias is generally caused by the fully-overlapped long syntactic constituents. All in all, we utilize them to impose higher attentions over the distinctive constituents.

We experiment on six benchmark corpora, including LCQMC, BQ, QQP, MRPC, PAWS_{QQP} and $\text{PAWS}_{\text{Wiki}}$. Experimental results show that our method obtains significant improvements compared to the BERT baseline, and achieves either comparable or state-of-the-art performance on different corpora. Our auxiliary experiments demonstrate that the utilization of ChatGPT[1] for data augmentation further improve the effects of attentive representation learning over distinctive information. Our contributions are three-fold: (1) We propose RAM which contributes to the representation of distinctive constituents with higher attentions. (2) We propose SAL which updates learning rate adaptively. It is effective in reducing bias during training. (3) We prove that the joint use of RAM and SAL enhances the BERT-based STM model.

2 Related Work

STM is a basic task in Natural Language Processing. Models for such tasks usually take two text sequences as input and determine the semantic relation-

[1] https://chat.openai.com/.

ship between them. Traditional techniques have explored a variety of neural network models based on different aspects of consideration, such as CNN [17], RNN [3,16,16] and GCN [22]. Recently, large-scale pre-trained language models such as BERT [6], RoBERTa [10] have significantly boosted the performance of various natural language tasks by pre-training on massive text resources with high-efficiency self-supervised learning objectives. For STM tasks, PLMs can achieve state-of-the-art performance on generic semantic matching benchmarks [19,20] by adding only a simple classification layer and fine-tuning under the sequence classification paradigm.

Although PLMs achieve enormous success in multiple tasks, some defects of them are gradually exposed with the deepening of research. DABERT [21] points out that self-attention mechanism ignore modeling the semantic difference between sentence pairs, which may be caused by the architecture itself. To alleviate this problem, DABERT propose a new type of attention called difference attention which is combined with vanilla attention to better capture the differences between text pairs. PAWS [25] ameliorates the problem by adding some high word overlap samples to the original training dataset. All of these previous works have resulted in significant performance improvements.

3 Approach

First, we define the STM task in a formal way. Given two text sequences $S^a = \{\, c_1^a, c_2^a, \ldots, c_{l_a}^a \,\}$ and $S^b = \{\, c_1^b, c_2^b, \ldots, c_{l_b}^b \,\}$, the goal of STM is to learn a binary classifier $y = \mathcal{F}(S^a, S^b)$ to determine whether S^a and S^b are semantically identical. Here, c_i^a and c_j^b denote the i-th and j-th character or word in the corresponding sequence respectively, and l_a, l_b represent the sequence length.

For BERT encoder, it is typical to concatenate text sequences S^a and S^b, incorporating special tokens to create a consecutive sequence $S_{\text{input}} = \{\, c_{[C]}, S^a, c_{[S]}^a, S^b, c_{[S]}^b \,\}$. This sequence is then inputted into the PLM encoder:

$$[\mathbf{h}_{[C]}, \mathbf{H}^a, \mathbf{h}_{[S]}^a, \mathbf{H}^b, \mathbf{h}_{[S]}^b] = \text{BERT}(S_{\text{input}}), \tag{1}$$

where $c_{[C]}$ and $c_{[S]}$ are special tokens used to connect text pairs, and the final hidden state $\mathbf{h}_{[C]}$ is used as an aggregate representation of the whole sequence.

3.1 Reversed Attention Mechanism

The attention structure inherent in PLM models renders them susceptible to over-reliance on lexical similarities [21], potentially leading to neglect of distinctions between text pairs. However, this limitation can also be leveraged to advantage. Specifically, we can postulate that the disregarded portion corresponds to the distinctions between a sentence pair.

Based on this assumption, we design Reversed Attention Mechanism (RAM), which highlights the distinctive parts between sentence pairs by reversing the distribution of attention scores. Specifically, the attention score is calculated via

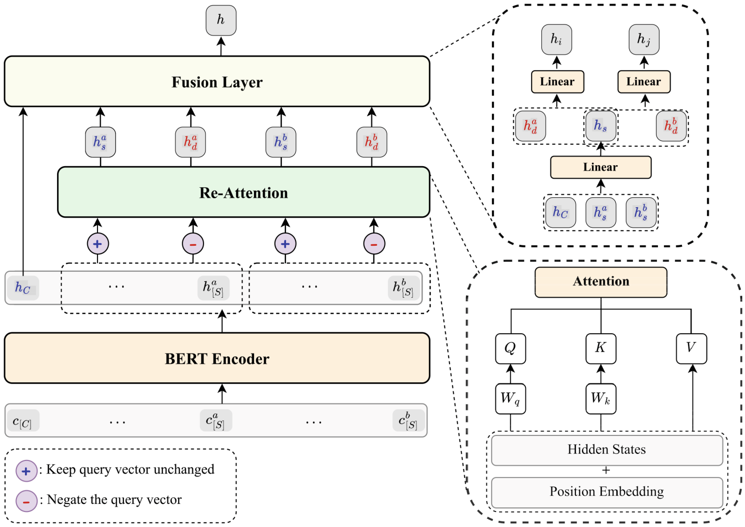


Fig. 2. The framework of our proposed model.

the dot product of the query and the corresponding set of keys. All we need is a simple operation, which is to negate the query vector. In this way, after Softmax function, we can just get the reversed attention distribution. The framework of RAM is shown in Fig. 2

Re-Attn. We design a simplified variant of attention named Re-Attn. Given the inputs $Q \in R^{m \times d_k}$, $K \in R^{n \times d_k}$ and $V \in R^{n \times d_v}$, the standard scaled dot-product attention in Transformer [18] can be described as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{2}$$

where m and n are the number of queries and key-value pairs. d_k represents the dimension of queries and keys, and d_v represents the dimension of values.

Inspired by multi-head attention in Transformer [18], we define Re-Attn which can be formalized as follows:

$$\text{Re-Attn}(q, H) = \text{Attention}(qW_q, HW_k, H), \tag{3}$$

where $H \in R^{n \times d}$ is a sequence of n vectors over which we would like to perform attention and $q \in R^d$ is a query vector. $W_q, W_k \in R^{d \times d_k}$ are trainable parameters. In contrast to multi-head attention, we discard the multi-head mechanism and do not project H with $W_v \in R^{d \times d_v}$. The changes we implement aim to preserve the output H of the encoder. This is because the context-aware representations H encoded by BERT are enriched with valuable semantic information, and we argue that these representations could potentially be compromised if we project them with randomly initialized W_v.

As the model's reliance on lexical similarities between sentence pairs is high, it can be inferred that high attention scores will be concentrated on these similar words. In this scenario, we derive the reversed attention distribution by negating the query vector. The resulting distribution will be focused on distinctions. Specifically, we incorporate position embeddings into the matrix H and subsequently employ Re-Attn to it using the following approach:

$$\begin{aligned}
\mathbf{h}_s^a &= \text{Re-Attn}(\mathbf{h}_{[S]}^a, \mathbf{H}^a \oplus \mathbf{h}_{[S]}^a),\\
\mathbf{h}_d^a &= \text{Re-Attn}(-\mathbf{h}_{[S]}^a, \mathbf{H}^a \oplus \mathbf{h}_{[S]}^a),\\
\mathbf{h}_s^b &= \text{Re-Attn}(\mathbf{h}_{[S]}^b, \mathbf{H}^b \oplus \mathbf{h}_{[S]}^b),\\
\mathbf{h}_d^b &= \text{Re-Attn}(-\mathbf{h}_{[S]}^b, \mathbf{H}^b \oplus \mathbf{h}_{[S]}^b),
\end{aligned} \tag{4}$$

where $\oplus$ denotes the concatenation operation and the position embedding is the sinusoidal positional embedding used in Transformer [18]. We define h_s^a, h_s^b as the features obtained by normal attention, and h_d^a, h_d^b as the features containing more distinctive information obtained by reversed attention.

Fusion. After obtaining all the features above, we fuse all features obtained from normal attention, h_s^a, h_s^b and $h_{[C]}$, into h_s at first:

$$h_s = \tanh(W_s(h_{[C]} \oplus h_s^a \oplus h_s^b) + b_s), \tag{5}$$

where $W_s \in R^{d\times 3d}, b_s \in R^d$ are weights and bias. h_s can be regarded as the feature containing global similarity information. Subsequently, we make h_s interact with h_d^a, h_d^b respectively. This step aims to capture and fuse the similarity and distinctive information. Then, We simply concatenate them as the input of classifier. The interaction process could be described as follows:

$$\begin{aligned}
h_i &= \text{gelu}(W_d(h_s \oplus h_d^a) + b_d),\\
h_j &= \text{gelu}(W_d(h_s \oplus h_d^b) + b_d),\\
h &= h_i \oplus h_j,
\end{aligned} \tag{6}$$

where $W_d \in R^{d\times 2d}, b_d \in R^d$ are parameters. Eventually, we leverage a two-layer feed forward network to make the final classification.

3.2 Sample-Based Adaptive Learning

In our study, we employ the ROUGE-L metric to assess the degree of similarity between pairs of texts. Based on our observations, we classify samples with a ROUGE-L score exceeding 0.75 as those exhibiting a high degree of lexical overlap. Conversely, we classify samples with a score below 0.4 as those displaying low lexical overlap. Through statistical analysis of the LCQMC training set, we discover that the proportion of positively-labeled to negatively-labeled samples exhibiting high lexical overlap is approximately 85:15, whereas the corresponding

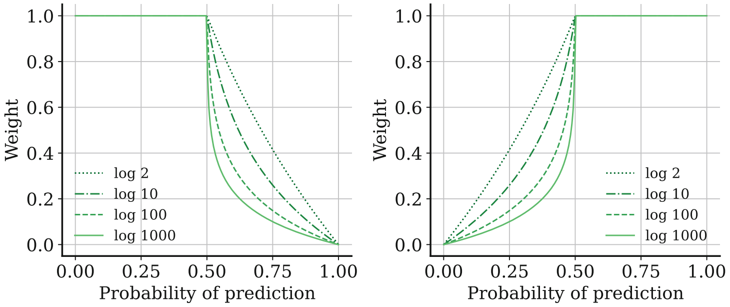


Fig. 3. The horizontal axis represents the probability predicted by the model, and the vertical axis represents the corresponding weight. When the ground truth is positive, the weight changes as shown in the left plot, otherwise as shown in the right plot.

ratio for low lexical overlap is around 9:91. This disparity could potentially introduce bias during the initial stages of model training. It is worth noting that a mere 13% of all negative samples exhibit high lexical overlap, which could render it challenging to rectify this bias in subsequent training.

To mitigate this issue, we propose a technique called Sample-based Adaptive Learning (SAL), which enhances the model's ability to focus on challenging samples through the use of distinct learning rates assigned at a fine-grained level. Specifically, we impose a penalty factor on the loss of samples that are already correctly classified during the training phase. Typically, the loss of a batch is calculated using the following formula:

$$loss = \frac{1}{n}\sum_{i=0}^{n} l_i, \tag{7}$$

where n is the batch size and l_i is the loss of the i^{th} sample in the batch.

We assign a weight to the loss of each sample in a batch. This weight is determined by the difference between the probability predicted by our model and the actual ground truth. Specifically, the weight is calculated as follows:

$$w_i = \begin{cases} \log_\alpha[1 - 2 \cdot |y_i - \hat{y_i}| \cdot (1 - \frac{1}{\alpha})], & \text{if } \delta(\hat{y_i}) = y_i \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

where α is a hyperparameter representing the base of the logarithm. $\hat{y_i}$, y_i are probability and ground truth respectively. $\delta(\hat{y_i}) = 1$ if $\hat{y_i} \geq 0.5$ else $\delta(\hat{y_i}) = 0$. Figure 3 depicts the correlation between the probability and corresponding weights across varying configurations of hyperparameter α.

Upon determining the weight of the loss for each individual sample, the aggregate loss of a batch can be computed utilizing the following formula:

$$loss = \frac{1}{n}\sum_{i=0}^{n} w_i l_i, \tag{9}$$

Table 1. Statistics of datasets.

Dataset	Train	Dev	Test
LCQMC	238,766	8,802	12,500
BQ	100,000	10,000	10,000
QQP	363,846	40,430	-
MRPC	3,668	408	1,725
PAWS_{QQP}	11,988	677	-
$\text{PAWS}_{\text{Wiki}}$	49,401	8,000	8,000

where n is the batch size, and w_i, l_i respectively represent the weight and loss of the i^{th} sample within the batch. It is noteworthy that the forward propagation process does not incorporate the computation of w_i, indicating that this aspect of the calculation is excluded from the gradient calculation during optimization. Our objective is to make the role of w_i similar to learning rate.

4 Experiments

4.1 Datasets

We evaluate our model on six STM datasets. The detailed information are listed in Table 1. Following are some basic descriptions of these datasets:

LCQMC [9] and BQ [1] are two Chinese datasets. The former is a large-scale open-domain corpus for matching Chinese questions, while the latter is a domain-specific corpus for matching bank-related questions.

The two English datasets are QQP [13] and MRPC [7]. Both of them are corpora of sentence pairs automatically extracted from online websites.

For the validation experiment, we select PAWS_{QQP} and $\text{PAWS}_{\text{Wiki}}$ [25]. Both contain extremely high lexical overlap samples constructed by applying word swapping and back translation to sentences from QQP and Wikipedia articles.

4.2 Implementation Details

The models employed in our experimentation are sourced exclusively from Hugging Face[2]. We apply AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$) with a weight decay rate of 0.01 and a warmup of 0.1 for learning rate decay. We tune the learning rate within the range of {2e-5, 3e-5, 5e-5}. The hyperparameter α in SAL is set to be 2 on QQP and 1000 on other datasets. For the MRPC dataset, we use an epoch of 10 and a batch size of 16. For the validation and ChatGPT experiments, we use an epoch of 5 and a batch size of 16. For all other datasets, we train models using an epoch of 5 and a batch size of 32. To ensure the statistical significance of the experimental results, we conduct each experiment five times using different seeds and report the average scores. The experiments on all datasets are performed utilizing an RTX 2080Ti GPU.

[2] https://huggingface.co/.

Table 2. Experimental results on two Chinese datasets. We report the average scores on their respective test sets and the best scores are depicted in brackets. Methods with † indicate the results from their papers, while methods with ‡ indicate our implementation.

Model	LCQMC		BQ	
	ACC.	F1	ACC.	F1
BERT‡ [6]	86.61	87.61	84.70	84.20
BERT-wwm† [5]	86.80	87.78	84.89	84.29
BERT-wwm-ext† [5]	86.68	87.71	84.71	83.94
ERNIE† [14]	87.04	88.06	84.67	84.20
MacBERT† [4]	87.00	-	85.20	-
ChineseBERT† [15]	87.40	-	85.20	-
LET-BERT† [11]	88.38	88.85	85.30	84.98
CBM-BERT† [2]	88.80	89.10	**86.16**	**87.44**
ours‡	88.72 (**89.13**)	89.07 (**89.34**)	85.30 (85.42)	84.82 (84.87)

Table 3. Experimental results on two English datasets. We report the accuracy scores on QQP development set and MRPC test set.

Model	QQP	MRPC
BERT† [6]	90.9	83.30
-large version	91.0	85.9
SS-BERT† [26]	91.4	-
R^2-Net† [24]	**91.6**	84.3
DC-Match† [27]	91.2	83.8
DABERT† [21]	91.3	-
Ours‡	91.4 (91.5)	**84.7** (**85.4**)

5 Results and Analysis

5.1 Main Results

The main results of comparison models on the Chinese and English dataset are presented separately in Table 2 and Table 3. Following previous works [11,27], we assess the matching performance using Accuracy and F1 for LCQMC and BQ, and Accuracy for QQP and MRPC. As for validation experiments, we follow the work [12] leveraging the F1 metric.

For Chinese PLMs, BERT-based models include six baselines. BERT is the official Chinese BERT model released by Google. Other subsequent models are enhanced during the pre-training stage to cater to the Chinese language. We report the results of our model that uses BERT as the backbone PLM.

Table 4. Experimental results on two PAWS datasets.

Model	PAWS_{QQP}	$\text{PAWS}_{\text{Wiki}}$
BERT	82.54	91.23
+RAM	83.79	91.67
+SAL	**84.47**	**92.08**

Table 5. Results of component ablation experiment on LCQMC test dataset.

Model	ACC.	F1
ours	**88.72**	**89.07**
w/o RAM	87.54	88.31
w/o SAL	87.82	88.47

In addition, we enumerate some of the state-of-the-art models that have emerged in recent years, namely LET-BERT [11] and CBM-BERT [2]. LET-BERT employs an external knowledge base known as HowNet to extract semantic information from words. This information is then integrated with BERT-encoded contextual information through a semantic-aware graph transformer. CBM-BERT enriches the semantic representation of a short text by incorporating additional context obtained by crawling the search engine results.

Table 2 shows that the five variants of BERT all surpass the original BERT, while our model outperforms all these BERT-based models. Furthermore, even if we do not introduce external knowledge or additional contextual information, our model outperforms LET-BERT on LCQMC and achieves comparable performance to CBM-BERT. On BQ dataset, there is a significant performance gap between the CBM-BERT and all other models, which may be attributed to the added contextual information that enables CBM-BERT to comprehend the semantics of a specific domain to a greater extent. Table 3 reveals that the extent of improvement in QQP is comparatively less significant than that in the Chinese dataset. This disparity could be attributed to the inadequacy of samples with high lexical overlap in the QQP corpus. To further verify the effectiveness of our method, we conduct validation experiments in Sect. 5.2.

5.2 Ablation and Verification Experiments

Verification Experiments. We conduct validation experiments on two datasets with high lexical overlap to further illustrate the effectiveness of our approach. The experimental results are shown in the Table 4. The initial row in Table 4 presents the performance of the baseline model. The subsequent row illustrates the performance of the model with our Re-Attn module applied on the baseline. Finally, the third row showcases the performance of the model after the further incorporation of our SAL module. The results indicate a steady improvement in performance on both datasets with the addition of each module. This further corroborates the impact of our method in enabling the model to effectively discern subtle differences between text pairs with high lexical overlap.

Ablation Experiments. We conduct ablation experiments on LCQMC to validate the effectiveness of each part in our method. The experimental results are demonstrated in the Table 5. It is evident from the results that the performance

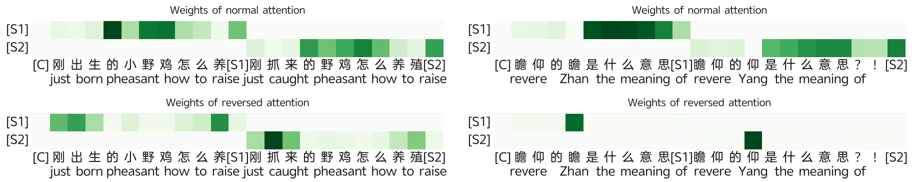


Fig. 4. Distributions of normal and reversed attention.

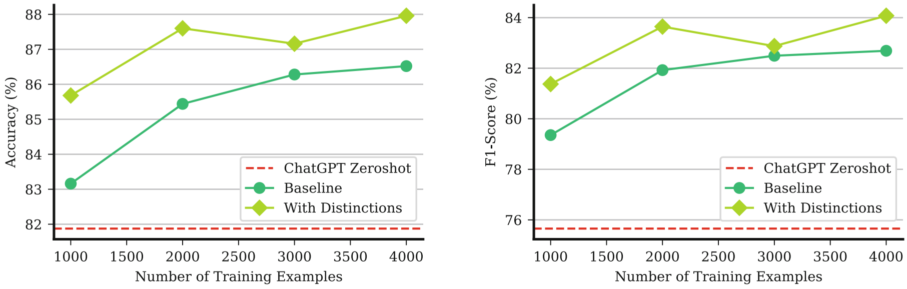


Fig. 5. We select RoBERTa as baseline. The result are the average of 5 runs.

degrades upon the removal of either of the two modules. In conjunction with the experimental findings presented in Table 4, it is apparent that our proposed modules exhibit complementary characteristics, resulting in superior performance when stacked together.

5.3 Case Study

To visualize how Re-Attn module works, We use two samples in LCQMC to analyze the attention distribution and the results are shown in Fig. 4. We only display the distribution of tokens which are used as queries in RAM. Normal attention weight distribution uses a positive query vector, while reversed attention uses a negative query vector. The texts on the left of Fig. 4 differ significantly in terms of the words ‘born’ and ‘caught’, and the reversed attention exactly assigns high attention scores to them. In Chinese, ‘revere’ is written as two characters ‘Zhan’ and ‘Yang’, and the two texts on the right of Fig. 4 inquire into the individual meanings of these characters. Similarly, ‘Zhan’ and ‘Yang’ acquire high attention score. Both results reveal that while normal attention focuses on similar words in a given text pair, reversed attention captures the differing portions of the text pair.

5.4 Effectiveness of Distinctive Information

Since LLMs like ChatGPT can provide high-quality responses to user’s questions, we utilize ChatGPT to generate the textual descriptions of distinctions between question pairs in QQP. The following is an example. User: “What are the differences between the following two questions? ‘What kind of first jobs do aviation majors tend to get?’ ‘What kind of first jobs do neuroscience majors tend to get?’” ChatGPT: “The first question is asking about the first jobs typically obtained by individuals with a major in aviation, while the second question is asking about the first jobs typically obtained by individuals with a major in neuroscience. The difference is in the specific field of study being referenced.”

Given the question S^a, S^b and the textual description of distinctions D^{ab}, the formulated inputs are structured according to the following prompt:

$$\begin{aligned}\text{input}_{\text{Q}} &= \{\,\text{The first question: } {<}S^a{>} \text{ The second question: } {<}S^b{>}\,\}\,,\\ \text{input}_{\text{D}} &= \{\,\text{The differences of the two questions: } {<}D^{ab}{>}\,\}\,.\end{aligned} \tag{10}$$

The input_{Q} and input_{D} would be encoded with individual encoders. Then, we unite the representations of two encoders for the final classification.

We randomly sample 4,000 and 500 pieces of data from the training and development sets, respectively. Subsequently, we train the model with varying amounts of data and compare the results against a baseline. The results are presented in Fig. 5. It is evident that incorporating distinctive information can aid in enhancing the model’s comprehension of semantic information.

6 Conclusion

In this work, we propose two novel approaches, namely RAM and SAL, aimed at addressing the challenge of distinguishing text pairs with high lexical overlap in the context of semantic matching tasks using PLMs. Despite their simplicity, both approaches are effective in enabling the model to better focus on and learn the distinctive information between text pairs. We also verified that providing literal descriptions of the distinctions between text pairs can enhance the model’s understanding of the matching task. This finding warrants further investigation.

Acknowledgements. The research is supported by National Key R&D Program of China (2020YFB1313601), National Science Foundation of China (62076174, 61836007).

References

1. Chen, J., Chen, Q., Liu, X., Yang, H., Lu, D., Tang, B.: The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In: EMNLP 2018, pp. 4946–4951 (2018)
2. Chen, M.Y., Jiang, H., Yang, Y.: Context enhanced short text matching using clickthrough data. CoRR abs/2203.01849 (2022)
3. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. In: Su, J., Carreras, X., Duh, K. (eds.) EMNLP 2016, pp. 551–561 (2016)
4. Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G.: Revisiting pre-trained models for Chinese natural language processing. In: EMNLP 2020. Findings of ACL, vol. EMNLP 2020, pp. 657–668 (2020)

5. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z.: Pre-training with whole word masking for Chinese BERT. IEEE ACM Trans. Audio Speech Lang. Process. **29**, 3504–3514 (2021)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019, pp. 4171–4186 (2019)
7. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: IWP@IJCNLP 2005 (2005)
8. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: EMNLP 2020, pp. 9119–9130 (2020)
9. Liu, X., et al.: LCQMC: A large-scale Chinese question matching corpus. In: COLING 2018, pp. 1952–1962 (2018)
10. Liu, Y., et al.: Roberta: A robustly optimized BERT pretraining approach abs/1907.11692 (2019)
11. Lyu, B., Chen, L., Zhu, S., Yu, K.: LET: linguistic knowledge enhanced graph transformer for chinese short text matching. In: AAAI 2021, pp. 13498–13506 (2021)
12. Peng, Q., Weir, D.J., Weeds, J., Chai, Y.: Predicate-argument based bi-encoder for paraphrase identification. In: ACL 2022, pp. 5579–5589 (2022)
13. Shankar Iyer, Nikhil Dandekar, K.C.: First quora dataset release: Question pairs (2012). https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs
14. Sun, Y., et al.: ERNIE: enhanced representation through knowledge integration abs/1904.09223 (2019)
15. Sun, Z., et al.: Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. In: ACL/IJCNLP 2021, pp. 2065–2075 (2021)
16. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: ACL 2015, pp. 1556–1566 (2015)
17. Tan, M., dos Santos, C.N., Xiang, B., Zhou, B.: Improved representation learning for question answer matching. In: ACL 2016 (2016)
18. Vaswani, A., et al.: Attention is all you need. NeurIPS **30** 5998–6008 (2017)
19. Wang, A., et al.: Superglue: A stickier benchmark for general-purpose language understanding systems. In: NeurIPS 2019, pp. 3261–3275 (2019)
20. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: ICLR 2019 (2019)
21. Wang, S., Liang, D., Song, J., Li, Y., Wu, W.: DABERT: dual attention enhanced BERT for semantic matching. In: COLING 2022, pp. 1645–1654 (2022)
22. Wu, L., Yang, Y., Zhang, K., Hong, R., Fu, Y., Wang, M.: Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In: SIGIR 2020, pp. 679–688 (2020)
23. Yu, L., Ettinger, A.: Assessing phrasal representation and composition in transformers. In: EMNLP 2020, pp. 4896–4907 (2020)
24. Zhang, Kun, Wu, Le., Lv, Guangyi, Wang, Meng, Chen, Enhong, Ruan, Shulan: Making the relation matters: relation of relation learning network for sentence semantic matching. Proceedings AAAI Conf. Artif. Intell. **35**(16), 14411–14419 (2021). https://doi.org/10.1609/aaai.v35i16.17694
25. Zhang, Y., Baldridge, J., He, L.: PAWS: paraphrase adversaries from word scrambling. In: NAACL-HLT 2019, pp. 1298–1308 (2019)

26. Zhao, Z., Zhang, Z., Hopfgartner, F.: SS-BERT: mitigating identity terms bias in toxic comment classification by utilising the notion of "subjectivity" and "identity terms". CoRR abs/2109.02691 (2021)
27. Zou, Y., et al.: Divide and conquer: Text semantic matching with disentangled keywords and intents. In: ACL 2022, pp. 3622–3632 (2022)

Oral: Information Extraction and Knowledge Graph

Reasoning Through Memorization: Nearest Neighbor Knowledge Graph Embeddings

Peng Wang, Xin Xie, Xiaohan Wang, and Ninyu Zhang(✉)

Zhejiang University, Hangzhou, China
{peng2001,xx2020,wangxh07,zhangningyu}@zju.edu.cn

Abstract. Previous knowledge graph embedding approaches usually map entities to representations and utilize score functions to predict the target entities, yet they typically struggle to reason rare or emerging unseen entities. In this paper, we propose *k*NN-KGE, a new knowledge graph embedding approach with pre-trained language models, by linearly interpolating its entity distribution with *k*-nearest neighbors. We compute the nearest neighbors based on the distance in the entity embedding space from the knowledge store. Our approach can allow rare or emerging entities to be memorized explicitly rather than implicitly in model parameters. Experimental results demonstrate that our approach can improve inductive and transductive link prediction results and yield better performance for low-resource settings with only a few triples, which might be easier to reason via explicit memory (Code is available at: https://github.com/zjunlp/KNN-KG).

Keywords: Pre-trained Language Models · Knowledge Graph Completion · Retrieval Augmentation

1 Introduction

Knowledge Graphs (KGs) organize facts in a structured way as triples in the form of `<subject, predicate, object>`, abridged as (s, p, o), where s and o denote entities and p builds relations between entities. Most KGs are far from complete due to emerging entities and their relations in real-world applications; hence KG completion—the problem of extending a KG with missing triples-has appeal to researchers [4–6,12,24,25,27].

Early, traditional KG completion methods, such as TransE [2], ComplEx [19], and RotatE [17], are knowledge embedding techniques that embed the entities and relations into a vector space and then obtain the predicted triples by leveraging a pre-defined scoring function to those vectors. Recently, another kind of KG embedding method utilize textual descriptions with language models [28], e.g., KG-BERT [22], StAR [20], and KGT5 [15], which are increasingly promising

P. Wang and X. Xie—Equal contribution.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 111–122, 2023.
https://doi.org/10.1007/978-3-031-44693-1_9

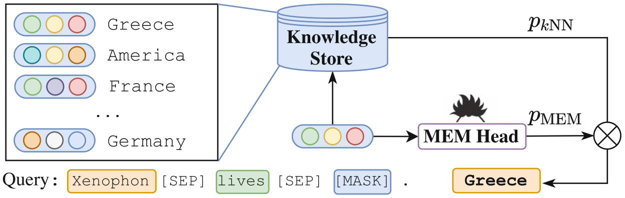


Fig. 1. Our kNN-KGE not only leverages entity prediction from softmax (MEM head in PLMs) but also retrieves the entities from the knowledge store constructed from entity descriptions and training triples.

techniques with empirical success. Note that those methods implicitly encode all of the relational knowledge in the weights of the parametric neural network via end-to-end training. However, a major limitation of these previous approaches is that they can hardly reason through rare entities evolving in a few triples or emerging entities unseen during training.

Note that human reasoning is facilitated by complex systems interacting together, for example, integration of current knowledge and retrieval from memory. Recent progress in memory-augmented neural networks has given rise to the design of modular architectures that separate computational processing and memory storage. Those memory-based approaches (or non/semi-parametric methods) have been applied to tasks such as language modeling [10] and question answering [9], which are expressive and adaptable.

Inspired by this, we propose kNN-KGE, an approach that extends knowledge graph embedding with language models by linearly interpolating its entity distribution with a k-nearest neighbors (kNN) model. As shown in Fig. 1, we construct a knowledge store of entities with pre-trained language models (PLMs) and retrieve nearest neighbors according to distance in the entity embedding space. Given a triple with a head or tail entity missing, we utilize the representation of `[MASK]` output as the predicted anchor entity embedding to find the nearest neighbor in the knowledge store and interpolate the nearest-neighbor distribution with the masked entity prediction. Thus, rare entities or emerging triples can be memorized explicitly, which makes reasoning through memorization rather than implicitly in model parameters. Experimental results on two datasets (FB15k-237 [18], and WN18RR [8]) in both transductive and inductive reasoning demonstrate the effectiveness of our approach. We further conduct a comprehensive empirical analysis to investigate the internal mechanism of kNN-KGE. Qualitatively, we observe that our approach is particularly beneficial for low-resource knowledge graph embedding, which might be easier to access via explicit memory. Our contributions can be summarized as follows:

- To the best of our knowledge, this is the first semi-parametric approach for knowledge graph embedding. Our work may open up new avenues for improving knowledge graph reasoning through explicit memory.
- We introduce kNN-KGE that can explicitly memorize rare or emerging entities, which is essential in practice since KGs are evolving.

- Experimental results on two benchmark datasets with transductive and inductive settings show that our model can yield better performance than baselines and is particularly beneficial for low-resource reasoning.

2 Methodology

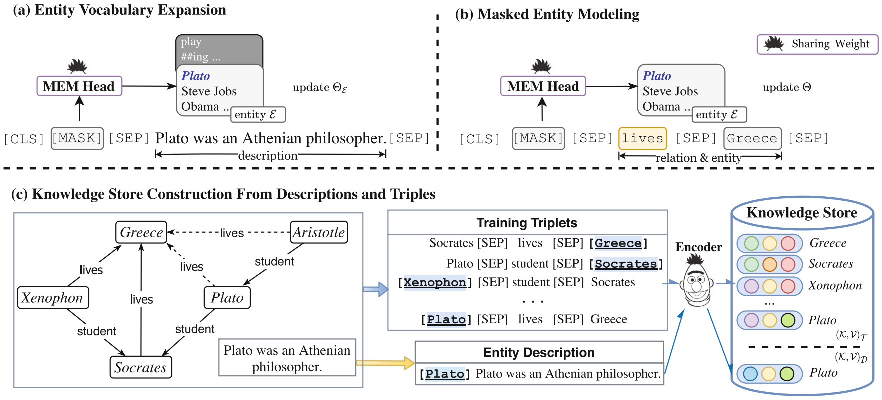

(a) Entity Vocabulary Expansion
play
##ing ...
Plato
Steve Jobs
Obama ..
entity $\mathcal{E}$
MEM Head
update $\Theta_{\mathcal{E}}$
[CLS] [MASK] [SEP] Plato was an Athenian philosopher. [SEP]
description
(b) Masked Entity Modeling
Sharing Weight
MEM Head
Plato
Steve Jobs
Obama ..
entity $\mathcal{E}$
update Θ
[CLS] [MASK] [SEP] lives [SEP] Greece [SEP]
relation & entity
(c) Knowledge Store Construction From Descriptions and Triples
Greece
lives
Aristotle
lives
lives
student
Xenophon
lives
Plato
student
student
Plato was an Athenian philosopher.
Socrates
Training Triplets
Socrates [SEP] lives [SEP] [Greece]
Plato [SEP] student [SEP] [Socrates]
[Xenophon] [SEP] student [SEP] Socrates
. . .
[Plato] [SEP] lives [SEP] Greece
Entity Description
[Plato] Plato was an Athenian philosopher.
Encoder
Knowledge Store
Greece
Socrates
Xonophon
...
Plato
$(\mathcal{K},\mathcal{V})_{\mathcal{T}}$
$(\mathcal{K},\mathcal{V})_{\mathcal{D}}$
Plato

Fig. 2. The virtual entity token embedding $\Theta_{\mathcal{E}}$ in the word embedding layer (head) is firstly optimized by Entity Vocabulary Expansion (Fig a). Then the model Θ shares the weight $\Theta_{\mathcal{E}}$ in Masked Entity Modeling (Fig b) for training. Finally, with the model trained above, the entities in triples and descriptions colored in blue will be encoded into contextualized entity representation and added to our knowledge store (Fig c). Examples are taken from [13]. (Color figure online)

2.1 Preliminary

Knowledge Graph. We define a knowledge graph with entity descriptions as a tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$, where $\mathcal{E}$ represents a set of entities, $\mathcal{R}$ represents relation types, $\mathcal{T}$ represents a set of triples and $\mathcal{D}$ represents the entity descriptions. For each triple in $\mathcal{T}$, it has the form (e_i, r_j, e_k) where $e_i, e_k \in \mathcal{E}$ is the head and tail entity respectively. For each entity $e_i \in \mathcal{E}$, there exists a text d_i to describe e_i. To complete missing triples in knowledge graphs, link prediction is proposed, which aims at predicting the tail entity given the head entity and the query relation, denoted by $(e_i, r, ?)$[1], where the answer is supposed to be always within the KG.

MEM Head. Like the "word embedding layer" (MLM Head) in the pre-trained language model that maps contextualized token representation into probability distribution of tokens in the vocabulary, MEM Head which consists of entity embeddings, maps contextualized entity representation to the probability distribution of the entity in the knowledge graph. We will illustrate the details of using MEM head in Sect. 2.5.

[1] or head entity prediction denoted by $(?, r, e_i)$.

2.2 Framework

In this Section, we introduce the general framework of the proposed approach as shown in Fig. 2. We first propose masked entity modeling and entity vocabulary expansion in Sect. 2.3, which converts link prediction into an entity prediction task. To address the issue of rare or unknown entities, in Sect. 2.4, we construct a knowledge store based on entity descriptions and triples in the training set to retrieve the entity by the anchor embedding (**the representation of `[MASK]` output**) during training. Lastly, in Sect. 2.5, we provide the details of inference, which makes reasoning through memorization rather than implicitly in model parameters.

2.3 Contextualized KG Representation

In this subsection, we treat the BERT model as the entity predictor because we convert the link prediction task to a masked entity modeling task, which uses the structural information or text description to predict the missing entity.

Masked Entity Modeling. For link prediction, given an incomplete triple $(e_i, r_j, ?)$, previous studies utilize KG embeddings or textual encoding to represent triple and leverage a pre-defined scoring function to those vectors. In this paper, we simply leverage masked entity modeling for link prediction, which makes the model predict the correct entity e_k like the Masked Language Model (MLM) task. The model only needs to predict the missing entity at the tail or head [26].

Specifically, given a querying triple $(e_i, r_j, ?)$ and the description d to the entity e_i, we concatenate this triple and the entity description d to obtain the input sequence x_k to predict the entity e_k as follows:

$$x_k = \texttt{[CLS]}\ e_i\ d\ \texttt{[SEP]}\ r_j\ \texttt{[SEP]}\ \texttt{[MASK]}\ \texttt{[SEP]}.$$

By masked entity modeling, the model can obtain the correct entity e_k by ranking the probability of each entity in the knowledge graph with p_{MEM}.

$$p_{\text{MEM}}(y|x) = p(\texttt{[MASK]} = e_k|x_k; \Theta), \tag{1}$$

where Θ represents the parameters of the pre-trained language models.

Note that the procedure of masked entity modeling is simple yet effective, and the one-pass inference speed is faster than the previous BERT-based model like StAR [20]. A detailed comparison of inference time can be found in Table 1. To do so, we use the same loss function to optimize our masked entity models.

$$\mathcal{L}_{\text{MEM}} = -\frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} 1_{e=e_k} \log p\left(\texttt{[MASK]} = e \mid x_k; \Theta\right), \tag{2}$$

where $|\mathcal{E}|$ is the number of total entities $\mathcal{E}$ and Θ represents the parameters of the model.

Table 1. Inference efficiency comparison. $|d|$ is the length of the entity description. $|\mathcal{E}|$, $|\mathcal{R}|$ and $|\mathcal{T}|$ are the numbers of all unique entities, relations and triples in the graph respectively. Usually, $|\mathcal{E}|$ exceeds hundreds of thousands and is much greater than $|\mathcal{R}|$.

Inference	Method	Complexity	Speed up	GPU time under RTX 3090
One Triple	StAR	$O\left(\lvert d\rvert^2(1+\lvert\mathcal{E}\rvert)\right)$	$\sim \lvert\mathcal{E}\rvert\times$	–
	*k*NN-KGE	$O\left(\lvert d\rvert^2+\lvert\mathcal{E}\rvert+\lvert\mathcal{T}\rvert\right)$		–
Entire Graph	StAR	$O\left(\lvert d\rvert^2\lvert\mathcal{E}\rvert(1+\lvert\mathcal{R}\rvert)\right)$	~2× because $\lvert\mathcal{E}\rvert\lvert\mathcal{R}\rvert \gg \lvert\mathcal{T}\rvert$	28 min
	*k*NN-KGE	$O\left((\lvert d\rvert^2+\lvert\mathcal{T}\rvert+\lvert\mathcal{E}\rvert)\lvert\mathcal{T}\rvert\right)$		15 min

We propose entity vocabulary expansion to utilize embeddings for each unique entity.

Entity Vocabulary Expansion. Since it is non-trivial to utilize subwords for entity inference, we directly utilize embeddings for each unique entity as common knowledge embedding methods [2] do. We represent the entities $e \in \mathcal{E}$ as **special tokens in language model's vocabulary**; thus, knowledge graph reasoning can reformulate as a masked entity prediction task as shown in Fig. 1 and Fig. 2.

$$x_d = \texttt{[CLS]}\ \text{prompt}(\texttt{[MASK]})\ \texttt{[SEP]}\ d\ \texttt{[SEP]}$$

We optimize those entity embeddings (random initialization) by predicting the entity e_i at the masked position with the other parameters fixed. Formally, we have the following:

$$\mathcal{L} = -\frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} 1_{e=e_i} \log p\left(\texttt{[MASK]} = e \mid x_d; \Theta\right) \tag{3}$$

where $|\mathcal{E}|$ is the number of total entities $\mathcal{E}$ and Θ represents the parameters of the model.

2.4 Knowledge Store

Inspired by recent progress in memory-augmented neural networks [9,10], we construct a knowledge store to explicitly memorize entities. We introduce the details of construction as follows:

From Descriptions $\mathcal{D}$. Let $f(\cdot)$ be the function that maps the entity e in the input x to a fixed-length vector representation computed by PLM. We use predesigned prompts to obtain entity embedding from entity descriptions. We can construct the knowledge store $(\mathcal{K}, \mathcal{V})_{\mathcal{D}}$ from descriptions (the set of descriptions $\mathcal{D}$ of all entities $\mathcal{E}$ in $\mathcal{G}$).

$$(\mathcal{K}, \mathcal{V})_{\mathcal{D}} = \{(f\left(x_d\right), e_i) \mid (d, e_i) \in \mathcal{G}\} \tag{4}$$

From Triples $\mathcal{T}$. Since different relations focus on different aspects of the same entity, it is intuitive to utilize different triples to represent entities. For example, given the triple (`Plato`, `lives`, ?), the model can reason through the triple of (`Plato`, `nationality`, `Greece`) in KGs to obtain `Greece`. Thus, we also construct a knowledge store from triples. We can construct the knowledge store $(\mathcal{K}, \mathcal{V})_{\mathcal{T}}$ from triples (the set of all triples $\mathcal{T}$ in $\mathcal{G}$).

$$(\mathcal{K}, \mathcal{V})_{\mathcal{T}} = \{(f(x_t), e_i) \mid (t, e_i) \in \mathcal{G}\} \tag{5}$$

2.5 Memorized Inference

In this Section, we will introduce memorized inference, the procedure of using KNN to choose the nearest embedding in the knowledge store.

Specifically, given a triple with the head or tail entity missing, we use the representation of `[MASK]` output as the predicted anchor entity embedding to find the nearest neighbor in the knowledge store.

$$\mathrm{d}(\mathbf{h}_i, \mathbf{h}_j) = |\mathbf{h}_i, \mathbf{h}_j|_2, \tag{6}$$

where $||_2$ refers to the Euclidean distance. $\mathbf{h}_{\texttt{[MASK]}}$ is the representation of the `[MASK]` token in the input sequence. $\mathbf{h_i}$ and $\mathbf{h_j}$ refers to the representation of two different embeddings (here, we refer to the $\mathbf{h}_{\texttt{[MASK]}}$ and the embedding in the knowledge store, respectively). Thus, we can obtain the probability distribution over neighbors based on a softmax of $k-$ nearest neighbors.

For each entity retrieved from the knowledge store, we choose only one nearest embedding in the knowledge store to represent the entity.

$$p_{\mathrm{kNN}}(y \mid x) \propto \sum_{(k_i, v_i) \in \mathcal{N}} 1_{y=v_i} \exp\left(-\mathrm{d}\left(k_i, f(x)\right)\right) \tag{7}$$

where x refers to the input sequence, and y refers to the target entity distribution by KNN via retrieving from the knowledge store. We interpolate the nearest neighbor distribution $p_{k\mathrm{NN}}$ with the model entity prediction p_{MEM} which can be seen in Eq. (1) and using a hyper-parameter λ to produce the final kNN-KGE distribution:

$$p(y \mid x) = \lambda p_{k\mathrm{NN}}(y \mid x) + (1-\lambda) p_{\mathrm{MEM}}(y \mid x) \tag{8}$$

3 Experiments

We conduct extensive experiments to evaluate the performance of kNN-KGE by answering the following research questions:

- **RQ1:** How does our kNN-KGE perform when competing with different types of transductive knowledge graph completion methods?
- **RQ2:** What is the benefits of our kNN-KGE when comparing with different inductive knowledge graph completion approaches?
- **RQ3:** How effective is our kNN-KGE in reasoning with less training or long-tailed triples?
- **RQ4:** How do different key modules in our kNN-KGE contribute to the overall performance?

Table 2. Transductive link prediction results on WN18RR and FB15k-237. *k*NN-KGE (w/o KS) represents we only use masked entity modeling without knowledge store. The **bold** numbers denote the best results in each genre while the underlined ones are the second-best performance.

Method	WN18RR					FB15k-237				
	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR
Graph embedding approach										
TransE [2] ◇	0.043	0.441	0.532	2300	0.243	0.198	0.376	0.441	323	0.279
DistMult [21] ◇	0.412	0.470	0.504	7000	0.444	0.199	0.301	0.446	512	0.281
R-GCN [16]	0.080	0.137	0.207	6700	0.123	0.100	0.181	0.300	600	0.164
ComplEx [19] ◇	0.409	0.469	0.530	7882	0.449	0.194	0.297	0.450	546	0.278
RotatE [17]	0.428	0.492	0.571	3340	0.476	0.241	0.375	0.533	177	0.338
TuckER [1]	0.443	0.482	0.526	–	0.470	0.226	0.394	0.544	–	0.358
ATTH [3]	0.443	0.499	0.573	–	0.486	0.252	0.384	0.549	–	0.348
Textual encoding approach										
KG–BERT [22]	0.041	0.302	0.524	97	0.216	–	–	0.420	153	–
MTL-KGC [11]	0.203	0.383	0.597	–	0.331	0.172	0.298	0.458	–	0.267
StAR [20]	0.243	0.491	**0.709**	**51**	0.401	0.205	0.322	0.482	**117**	0.296
KGT5 [15]	0.487	–	0.544	–	0.508	0.210	–	0.414	–	0.276
*k*NN-KGE (w/o KS)	0.399	0.527	0.633	1365	0.481	0.269	0.396	0.547	283	0.360
*k*NN-KGE	**0.525**	**0.604**	0.683	986	**0.579**	**0.280**	**0.404**	**0.550**	185	**0.370**

3.1 Experimental Settings

We evaluate our method on FB15k-237 [18] and WN18RR [8] in both transductive and inductive settings, which are widely used in the link prediction literature.

3.2 Transductive Experiments (RQ1)

As shown in Table 2, our proposed *k*NN-KGE can obtain state-of-the-art or competitive performance on all these datasets, especially significant improvement

Table 3. Inductive link prediction results on WN18RR and FB15k-237 in inductive setting. †Resulting numbers are reported by [7]. The **bold** numbers denote the best results in each genre while the underlined ones are the second-best performance.

Method	WN18RR ind				FB15k-237 ind			
	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
BE-BOW†	0.045	0.244	0.450	0.180	0.103	0.184	0.316	0.173
BE-DKRL†	0.031	0.141	0.282	0.139	0.084	0.151	0.263	0.144
BLP-TransE†	0.135	**0.361**	**0.580**	0.285	0.113	0.213	**0.363**	0.195
BLP-DistMult†	0.135	0.288	0.481	0.248	0.076	0.156	0.286	0.146
BLP-ComplEx†	0.156	0.297	0.472	0.261	0.081	0.154	0.283	0.148
*k*NN-KGE	**0.223**	0.320	0.431	**0.294**	**0.146**	**0.214**	0.293	**0.198**

(0.443 → 0.525 on WN18RR and 0.252 → 0.280 on FB15k-237) in terms of Hits@1. We hold that the improvement is mainly attributed to better retrieval by our masked entity modeling and knowledge store.

3.3 Inductive Experiments (RQ2)

From Table 3, we observe that our *k*NN-KGE method can yield better or comparable performance compared with the previous state-of-the-art method, which demonstrates the effectiveness of the knowledge store. We notice that *k*NN-KGE reaches the best Hits@1, which further validates the advantages of reasoning through memorization.

3.4 Results with Less Training and Long-Tailed Triples (RQ3)

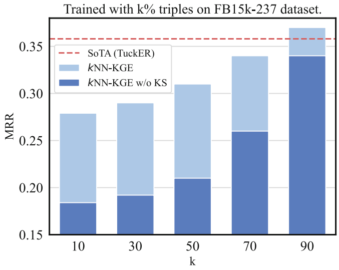


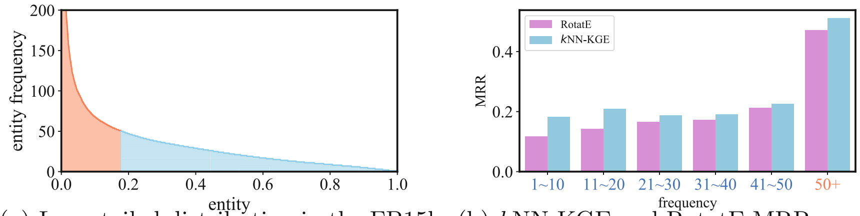
Fig. 3. Varying the size of the training samples. In the low resource setting, knowledge store monotonically improves performance.

As shown in Sect. 3.2 and 3.3, retrieving neighbors from the knowledge store can improve the model's performance. When facing emerging entities, the translation-based model (such as TransE) have to add new entities or new triples to the models and retrain the entire model, which requires huge computation resource. We conduct an ablation study compared with the state-of-the-art model TuckER [1] (red line) and our approach without retrieving knowledge store (w/o KS). Moreover, we notice that our approach with 70% samples can yield comparable performance with the previous SoTA model in MRR (Fig. 3).

Improvements in Tail Entities. We find that those tail entities in long-tailed distributions can achieve performance improvement. We can observe from Fig. 4 that only 18% entities show up more than 50 times and *k*NN-KGE obtain improvements (0.118 → 0.183) with the entities occurring less than 20 times.

3.5 Analysis (RQ4)

In this Section, we first conduct a case study and show the improvements in tail entities by the knowledge store. Second, we visualize the entity embeddings in the knowledge store with a query triple. Then, we analyze the effect of hyperparameters on retrieving the entities in the knowledge store. Lastly, we summarize the size of the knowledge store and the speed of reasoning on two datasets: WN18RR and FB15k-237.



(a) Long-tailed distribution in the FB15k-237. 82% entities occur < **50 times**.

(b) *k*NN-KGE and RotatE MRR result in FB15k-237 dataset.

Fig. 4. Significant improvement in entities with frequency below 50 colored in blue. (Color figure online)

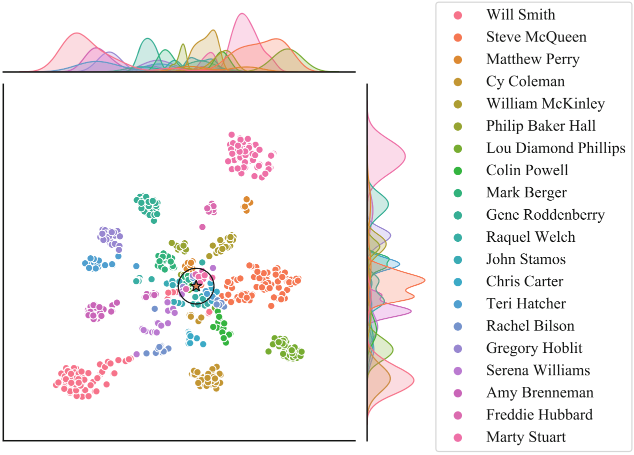


Fig. 5. 2D t-SNE visualisation of 20 nearest neighbor entities with all their embeddings in knowledge store retrieved by predicting the missing entity in triple (`?`, `nationality`, `America`). Anchor embedding is marked as $\star$.

Case Study. We also conduct case studies to analyze the different reasoning results with or without querying the knowledge store. From Table 4, we observe that *k*NN-KGE w/o Knowledge Store can infer better entities given head/tail entities and relations, further demonstrating the effectiveness of the proposed approach. Note that *k*NN-KGE can explicitly memorize those entities in Knowledge Store; thus, it can directly reason through memorization.

Table 4. First five entities with their probability predicted by kNN-KGE w/o Knowledge Store, and its reranking with kNN-KGE, for two example queries.

Query: `(?,ethnicity,Timothy Spall)`		
Rank	kNN-KGE w/o Knowledge Store	kNN-KGE
1	Jewish people (0.285)	**English people** (0.548)
2	**English people** (0.202)	Jewish people (0.028)
3	British people (0.132)	British people (0.026)
4	White British (0.080)	Irish people in ... (0.021)
5	Scottish people (0.037)	Irish people (0.018)

Visualization of Entity Embeddings in Knowledge Store. Since we build our knowledge store with different entity embeddings from different aspects of the same entity, we visualize (Fig. 5) those embeddings in the knowledge store. Specifically, we random sample an input triple (`?`, `nationality`, `America`), retrieve and visualize the 20 nearest neighbor entities with their embeddings in the knowledge store.

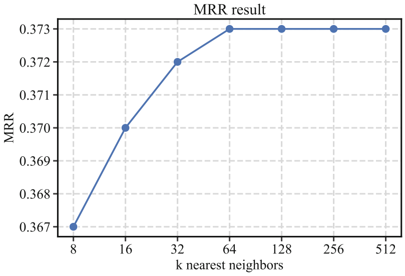


Fig. 6. Effect of the number of neighbors from knowledge store.

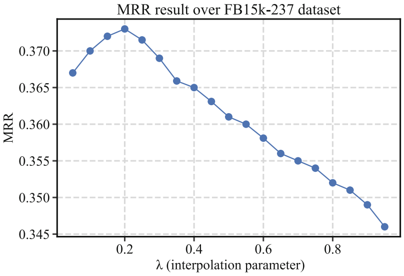


Fig. 7. Effect of interpolation parameter λ on FB15k-237 dataset.

Impact of Number of Neighbors. To validate the impact of the number of neighbors, we conduct experiments with the different numbers of k nearest neighbor entities to retrieve from the knowledge store.

From Fig. 6, we find that the model performance continues to improve as K increases until it converges when reaching a threshold ($k = 64$). We think this is because those entities retrieved from the knowledge store are far from the anchor embedding, thus, having a low influence on the knowledge graph reasoning results.

Impact of Interpolation. Since we use a parameter λ to interpolate between the BERT model distribution and the retrieved distribution from the kNN search over the dataset, we further conduct experiments to analyze the interpolation.

From Fig. 7, we notice that $\lambda = 0.2$ is optimal on the FB15k-237 dataset. The suitable λ can help correct the model from inferring the wrong entities, which can be seen in Table 4.

4 Conclusion and Future Work

In this paper, we propose a novel semi-parametric approach for knowledge graph embedding dubbed kNN-KGE, which can outperform previous knowledge graph embedding models in both transductive and inductive settings by directly querying entities at test time. The success of kNN-KGE suggests that explicitly memorizing entities can be helpful for knowledge graph reasoning [14,26].

In the future, we plan to 1) improve the efficiency of the kNN-KGE with smaller knowledge store and faster inference speed; 2) explore how to edit [23] and delete entities from the knowledge store for dynamic KG reasoning; 3) apply our approach to other tasks, such as question answering and fact verification.

References

1. Balazevic, I., Allen, C., Hospedales, T.M.: Tucker: tensor factorization for knowledge graph completion. In: Proceeding of EMNLP (2019)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NeurIPS (2013)
3. Chami, I., Wolf, A., Juan, D., Sala, F., Ravi, S., Ré, C.: Low-dimensional hyperbolic knowledge graph embeddings. In: Proceedings of ACL (2020)
4. Chen, M., et al.: Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23–29 July 2022, pp. 1966–1972. ijcai.org (2022). https://doi.org/10.24963/ijcai.2022/273
5. Chen, M., et al.: Meta-knowledge transfer for inductive knowledge graph embedding. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022, pp. 927–937. ACM (2022). https://doi.org/10.1145/3477495.3531757
6. Chen, X., et al.: KnowPrompt: knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In: Proceedings of the ACM Web Conference 2022. ACM (2022). https://doi.org/10.1145/3485447.3511998
7. Daza, D., Cochez, M., Groth, P.: Inductive entity representations from text via link prediction. In: Proceedings of WWW (2021)
8. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of AAAI (2018)
9. Kassner, N., Schütze, H.: BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In: Findings of EMNLP (2020)

10. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: Nearest neighbor language models. In: Proceedings of ICLR (2020)
11. Kim, B., Hong, T., Ko, Y., Seo, J.: Multi-task learning for knowledge graph completion with pre-trained language models. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1737–1743. International Committee on Computational Linguistics, Barcelona (2020). https://doi.org/10.18653/v1/2020.coling-main.153, https://aclanthology.org/2020.coling-main.153
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of AAAI (2015)
13. Liu, S., Grau, B., Horrocks, I., Kostylev, E.: Indigo: GNN-based inductive knowledge graph completion using pair-wise encoding. In: Proceedings of NeurIPS (2021)
14. Qiao, S., et al.: Reasoning with language model prompting: a survey (2023)
15. Saxena, A., Kochsiek, A., Gemulla, R.: Sequence-to-sequence knowledge graph completion and question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2814–2828. Association for Computational Linguistics, Dublin (2022). https://doi.org/10.18653/v1/2022.acl-long.201, https://aclanthology.org/2022.acl-long.201
16. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
17. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: ICLR (2019)
18. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of EMNLP (2015)
19. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML (2016)
20. Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., Chang, Y.: Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of WWW (2021)
21. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of ICLR (2015)
22. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. CoRR (2019)
23. Yao, Y., et al.: Editing large language models: problems, methods, and opportunities (2023)
24. Zhang, N., Deng, S., Sun, Z., Chen, J., Zhang, W., Chen, H.: Relation adversarial network for low resource knowledge graph completion. In: Proceedings of WWW (2020)
25. Zhang, N., et al.: AliCG: fine-grained and evolvable conceptual graph construction for semantic search at Alibaba. In: Proceedings of KDD (2021)
26. Zhang, N., Li, L., Chen, X., Liang, X., Deng, S., Chen, H.: Multimodal analogical reasoning over knowledge graphs (2023). https://openreview.net/forum?id=NRHajbzg8y0P
27. Zhang, N., Xie, X., Chen, X., Deng, S., Ye, H., Chen, H.: Knowledge collaborative fine-tuning for low-resource knowledge graph completion. J. Softw. **33**(10), 3531–3545 (2022)
28. Zhao, W.X., et al.: A survey of large language models (2023)

MACO: A Modality Adversarial and Contrastive Framework for Modality-Missing Multi-modal Knowledge Graph Completion

Yichi Zhang, Zhuo Chen, and Wen Zhang(✉)

Zhejiang University, Hangzhou, China
{zhangyichi2022,zhuo.chen,zhang.wen}@zju.edu.cn

Abstract. Recent years have seen significant advancements in multi-modal knowledge graph completion (MMKGC). MMKGC enhances knowledge graph completion (KGC) by integrating multi-modal entity information, thereby facilitating the discovery of unobserved triples in the large-scale knowledge graphs (KGs). Nevertheless, existing methods emphasize the design of elegant KGC models to facilitate modality interaction, neglecting the real-life problem of missing modalities in KGs. The missing modality information impedes modal interaction, consequently undermining the model's performance. In this paper, we propose a modality adversarial and contrastive framework (MACO) to solve the modality-missing problem in MMKGC. MACO trains a generator and discriminator adversarially to generate missing modality features that can be incorporated into the MMKGC model. Meanwhile, we design a cross-modal contrastive loss to improve the performance of the generator. Experiments on public benchmarks with further explorations demonstrate that MACO could achieve state-of-the-art results and serve as a versatile framework to bolster various MMKGC models.

Keywords: Multi-modal Knowledge Graph · Knowledge Graph Completion · Generative Adversarial Networks

1 Introduction

Knowledge graph completion (KGC) [1] is a popular research topic that focuses on discovering unobserved knowledge in knowledge graphs (KGs) [15], which consist of massive entities and relations in the form of triple *(head entity, relation, tail entity)*. Multi-modal information like images serve as the supplementary information for entities and could also benefit the KGC models, which is known as multi-modal KGC (MMKGC) [10,14,17] in the research community.

Typically, MMKGC is accomplished by embedding-based methods, which embed entities and relations in the KGs to a low-dimensional embedding space and design score functions to model the triple structure, thus learning what's

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 123–134, 2023.
https://doi.org/10.1007/978-3-031-44693-1_10

known as structural embeddings. Additionally, after feature extraction, multimodal information such as images needs to be fused and interacted with structural embeddings to improve KGC performance. This highlights the importance of the structural-visual modality interaction and fusion for achieving better MMKGC performance.

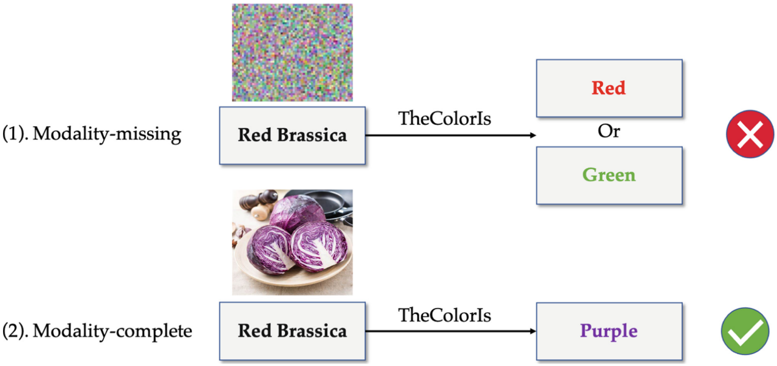


Fig. 1. A case of the influence of missing modality in KG. Without the help of the visual information, the color of red brassica might be predicted as red or green due to the contextual information of KG. The meaningful visual information could guide the KGC model to accurately predict the tail entity. (Color figure online)

However, construction of real-world KGs typically involves multiple heterogeneous data sources, making it challenging to guarantee complete modality information for all entities and resulting in the modality-missing problem in MMKGC. Such a problem would harm the modality interaction and lead to poor KGC performance. Though existing MMKGC methods [10,14,17] incorporate various approaches to align the structural and visual information, they tend to overlook the modality-missing problem. These methods usually apply simple solutions like random initialization to complete the missing visual information, which might introduce noise into the MMKGC model and loss of some crucial information. Figure 1 illustrates how meaningful visual information could improve the performance of KGC models, which also reflects the importance of completing the visual information of the entity.

To address the missing-modality problem, we propose a **M**odality **A**dversarial and **CO**ntrastive (MACO for short) framework for modality-missing MMKGC. Leveraging the generative adversarial framework [5], we integrate a pair of generator and discriminator to generate missing visual features conditioned on the entity structural information. Besides, we design a cross-modal contrastive loss [8] to enhance the quality of generated features and improve training stability [20]. The generated visual features would be used in the MMKGC models. To demonstrate the effectiveness of MACO, we conduct comprehensive experiments on the public benchmarks and make further explorations. Experimental results prove that MACO could achieve state-of-the-art (SOTA) KGC results compared with baseline methods and serve as a general enhancement framework for different MMKGC models.

The contributions of our work can be summarized as follows:

1. We are the first work dedicated to addressing the modality-missing problem in the MMKGC task.
2. We propose a novel framework MACO to generate realistic visual features and design cross-modal contrastive loss to improve the quality of the generated features.
3. We demonstrate the effectiveness of MACO with comprehensive experiments on public benchmarks with further exploration, which prove that MACO could achieve SOTA results in the modality-missing MMKGC.

2 Related Works

2.1 Multi-modal Knowledge Graph Completion

Knowledge graph completion (KGC) aims to discover the unobserved triples in the KGs. Knowledge graph embedding (KGE) [15] is a mainstream approach towards KGC. General KGE methods [1,11,13,19] embed the entities and relations of KGs into low-dimensional vector spaces and modeling the triple structure with different score functions.

As for multi-modal knowledge graph completion (MMKGC), the modal information (images, textual descriptions) should be considered in the embedding model. IKRL [17] projects the visual features into the same vector space of structural information and considers the visual features in the score function. TBKGC [10] further consider visual and textual information and make exploration about modal fusion. TransAE [16] employs an auto-encoder to encode the modal information better. RSME [14] design several gates to select the truly useful modal information. Recent methods like OTKGE [2] and MoSE [21] make further steps in multi-modal fusion.

2.2 Incomplete Multi-modal Learning

Incomplete multimodal learning (IML) has attracted extensive attention in the research community as the modality-missings situation is common in practice [4,6]. The mainstream solutions towards IML are divided into two categories: the generative methods and the joint learning methods. Generative methods are designed to learn the data distribution and generate the missing modality information with generative frameworks such as GAN [5] and VAE [7]. Joint learning methods, however, attempt to learn robust joint embeddings under missing modalities.

As for MMKGs, the modality-missing problem has long been neglected. As for the KGC task, existing methods usually ignore such a problem or just complete the missing information with naive approaches like random initialization. We believe that it is important to complete the missing entity modal information in the process of KGC, to enrich the KGs and improve the performance of KGC.

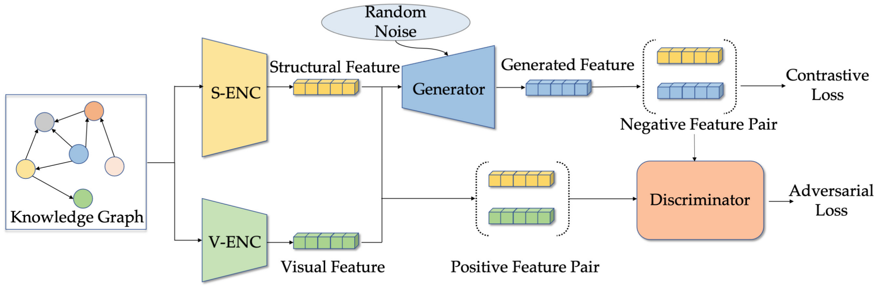


Fig. 2. The model architecture of MACO. There are three key designs of MACO: the feature encoders, the adversarial training, and the cross-modal contrastive loss. The structural encoder (S-ENC) and visual encoder (V-ENC) are used to capture the structural/visual features. The adversarial training would employ a generator and a discriminator and apply adversarial training. The cross-modal contrastive loss is designed to improve the quality of the generated features.

3 Methodology

3.1 Preliminary

A KG could be denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the entity set, relation set, triple set respectively. As for MMKG, the image set of each entity $e \in \mathcal{E}$ can be denoted as $\mathcal{I}(e)$, which could be $\varnothing$ when the modal information is missing. Furthermore, in the scenario of missing modality, we can partition the entity set into two disjoint parts $\mathcal{E}_c$ and $\mathcal{E}_m$, which include the modality-complete ($\mathcal{E}_c$) and modality-missing ($\mathcal{E}_m$) entities respectively.

3.2 MACO Framework

In this section, we will provide a comprehensive overview of our modality adversarial and contrastive framework (MACO) detailedly. A detailed illustration of MACO's model architecture can be found in Fig. 2. MACO is primarily characterized by three key components: feature encoders, modality-adversarial training, and cross-modal contrastive loss. The primary objective of our MACO framework is to complete the visual information of the modality-missing entities.

Feature Encoders. We have designed feature encoders to encode the features of different modalities in the knowledge graph (KG). Specifically, we apply a structural encoder $\mathbf{S}$ to encode the structural information of each entity in the KG, while employing a visual encoder $\mathbf{V}$ to encode the visual information of each entity. In our implementation, $\mathbf{S}$ is a L-layer relational graph convolution network (R-GCN) [9], which could capture the structural features in the KG. For each layer $l (l = 1, 2, \ldots, L)$, the structural features are updated by the message-passing process denoted as:

$$\boldsymbol{s}_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{|\mathcal{N}_i^r|}\mathbf{W}_r^{(l)}\boldsymbol{s}_j^{(l)} + \mathbf{W}_0^{(l)}\boldsymbol{s}_i^{(l)}\right) \tag{1}$$

where $\boldsymbol{s}_i$ is the structural feature of entity e_i, $\mathcal{N}_i^r$ is the neighbor set of e_i under relation $r \in \mathcal{R}$, σ is the ReLU activation function [9], $\mathbf{W}_0, \mathbf{W}_r$ are the learnable projection matrices.

Besides, we employ a vision transformer (ViT) [3] to capture the visual features of the entities $e_i \in \mathcal{E}_{comp}$. For those entities with more than one image, we apply to mean pooling to aggregate the visual features. The visual feature of entity e_i is denoted as $\boldsymbol{v}_i$.

Modality-Adversarial Training. The second key component of MACO is the modality-adversarial training, which includes a generator $\mathbf{G}$ and a discriminator $\mathbf{D}$. $\mathbf{G}$ is a conditional generator, aiming to generate the visual information given the structural feature of an entity. This design of the conditional generator is intended to enable the generator to produce visual features appropriate for the current entity. Hence, we also term $\mathbf{G}$ the modality-adversarial generator. We implement $\mathcal{G}$ with a two-layer feed-forward network (FFN), which could be denoted as:

$$\mathbf{G}(\boldsymbol{s}, z) = \mathbf{W}_2\left(\delta(\mathbf{W}_1[\boldsymbol{s}; z] + \mathbf{b}_1)\right) + \mathbf{b}_2 \tag{2}$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are the parameters of two feed-forward layers, δ is the LeakyReLU [18] activation function, $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the random noise, and $[;]$ is the concentrate operation. We denote $\boldsymbol{g}_i = \mathbf{G}(\boldsymbol{s}_i, z)$ as the generated visual feature for entity e_i.

Moreover, $\mathbf{D}$ serves as a classifier designed to discriminate whether a pair of structural feature $\boldsymbol{s}$ and visual features $\boldsymbol{v}$ are compatible, which would be a binary classifier. The existing structural-visual feature pair $(\boldsymbol{s}_i, \boldsymbol{v}_i)$ for $e_i \in \mathcal{E}_{comp}$ are the positive feature pairs with label 1, while the generated pair $(\boldsymbol{s}_i, \mathbf{G}(\boldsymbol{s}_i, z))$ are viewed as negative feature pairs with ground-truth label 0. In practice, $\mathbf{D}$ is another two-layer network denoted as:

$$\mathbf{D}(\boldsymbol{s}, \boldsymbol{v}) = \mathbf{W}_4[\delta(\mathbf{W}_3\boldsymbol{s} + \mathbf{b}_3); \boldsymbol{v}] + \mathbf{b}_4 \tag{3}$$

where $\mathbf{W}_3, \mathbf{W}_4, \mathbf{b}_3, \mathbf{b}_4$ are the parameters of the network.

During training, we apply binary cross-entropy as the loss function to optimize the models:

$$\mathcal{L}_{adv} = -\left(\frac{1}{|\mathcal{E}|}\sum_{e_i\in\mathcal{E}}\log(1 - \mathbf{D}(\boldsymbol{s}_i, \boldsymbol{g}_i)) + \frac{1}{|\mathcal{E}_c|}\sum_{e_i\in\mathcal{E}_c}\log(\mathbf{D}(\boldsymbol{s}_i, \boldsymbol{v}_i))\right) \tag{4}$$

In the adversarial context, the generator $\mathbf{G}$ aims to generate convincing visual features and fool the discriminator $\mathbf{D}$ while $\mathbf{D}$ is designed to make robust predictions to recognize those manually generated features. Thus, $\mathbf{G}$ and $\mathbf{D}$ would play a mini-max game and optimize their parameters in an adversarial manner, which could be denoted as:

$$\min_{\mathbf{D}} \max_{\mathbf{G}} \mathcal{L}_{adv} \tag{5}$$

Cross-Modal Contrastive Loss. In the mentioned design, we utilized the design concepts of generative adversarial networks (GANs) [5], however, the training of GAN models is unstable, and the quality of the generated features is difficult to control [8], potentially decreasing the generator's performance.

Thus, we propose another cross-modal contrastive module to contrast the structural features and the generated visual features, aiming to maximize their mutual information and improve the quality of the generated visual features. A pair of structural feature $\boldsymbol{s}_i$ and generated visual feature $\boldsymbol{g}_i$ of the same entity e_i is regarded as a positive pair and we apply in-batch negative sampling to construct negative pairs. The contrastive loss could be denoted as:

$$\mathcal{L}_{con} = -\frac{1}{|\mathcal{E}|}\sum_{e_i \in \mathcal{E}} \log \frac{\gamma(\boldsymbol{s}_i, \boldsymbol{g}_i)}{\gamma(\boldsymbol{s}_i, \boldsymbol{g}_i) + \sum_{e'_j \in \mathcal{N}(e_i)} \gamma(\boldsymbol{s}_i, \boldsymbol{g}'_j)} \tag{6}$$

where $\mathcal{N}(e_i)$ is the negative entity set of e_i, $\gamma(\boldsymbol{s}_i, \boldsymbol{g}_j)$ is the score of a structural-visual feature pair. The score is calculated as:

$$\gamma(\boldsymbol{s}_i, \boldsymbol{g}_j) = \exp\left(\cos(\boldsymbol{s}_i, \boldsymbol{g}_j)/\tau\right) \tag{7}$$

where cos is the cosine similary and τ is the temperature. In practice, we apply in-batch sampling [20] to get the negative entities. When training $\mathbf{G}$, the contrastive loss would be added to the overall objective to enhance the performance of $\mathbf{G}$. Thus, the overall training objective of MACO is:

$$\min_{\mathbf{D}} \max_{\mathbf{G}} \mathcal{L}_{adv} + \min_{\mathbf{G}} \alpha \mathcal{L}_{con} \tag{8}$$

where α is the coefficient of the contrastive loss.

3.3 Missing Modality Completion and Downstream Usage

Following the above design of MACO, we could obtain the generator $\mathbf{G}$ and a discriminator $\mathbf{D}$. The subsequent step is to complete the missing modality information with $\mathbf{G}$ and $\mathbf{D}$. In our design, for an entity e_i, we would first generate K visual features $\boldsymbol{g}_i$ by $\mathbf{G}$ and assess their compatibility with the structural feature $\boldsymbol{s}_i$ using $\mathbf{D}$. Then we apply to mean pooling to the valid visual feature $\boldsymbol{g}_i$ to obtain the final visual feature $\boldsymbol{v}_i$. This process can be denoted as $\boldsymbol{v}_i = \frac{\sum_{j=1}^{K} y_{i,j}\boldsymbol{g}_j}{\sum_{j=1}^{K} y_{i,j}}$: where $y_{i,j} \in \{0,1\}$ is the prediction result of $(\boldsymbol{s}_i, \boldsymbol{g}_j)$ made by $\mathbf{D}$. Further, we propose two strategies to complete the missing modality. The first is to generate only for those modality-missing entities in $\mathcal{E}_m$. The second is to generate for all the entities in $\mathcal{E}$ and change the original visual features for $e \in \mathcal{E}_c$. We name the two strategies as Gen and All-Gen respectively.

After generating the visual features, they will be used to initialize the visual embeddings of entities in the KGC model. A score function $\mathcal{S}(h, r, t)$ is designed to measure the triple plausibility, which would calculate the triple score with the structural and visual embeddings. To assign the positive triples with higher scores, we apply margin-rank loss [1] to train the KGC model, denoted as:

$$\mathcal{L}_{kgc} = \max\left(0, \lambda - \mathcal{S}(h, r, t) + \sum_{i=1}^{N} p_i \mathcal{S}(h'_i, r'_i, t'_i)\right) \tag{9}$$

where λ is the margin and p_i is the self-adversarial weight of the negative samples proposed by [11]. It is denoted as:

$$p_i = \frac{\exp(\beta \mathcal{S}(h'_i, r'_i, t'_i))}{\sum_{j=1}^{N} \exp(\beta \mathcal{S}(h'_j, r'_j, t'_j))} \tag{10}$$

where β is the temperature. During our experiments, we would try several different score functions to demonstrate the effectiveness of MACO.

4 Experiments

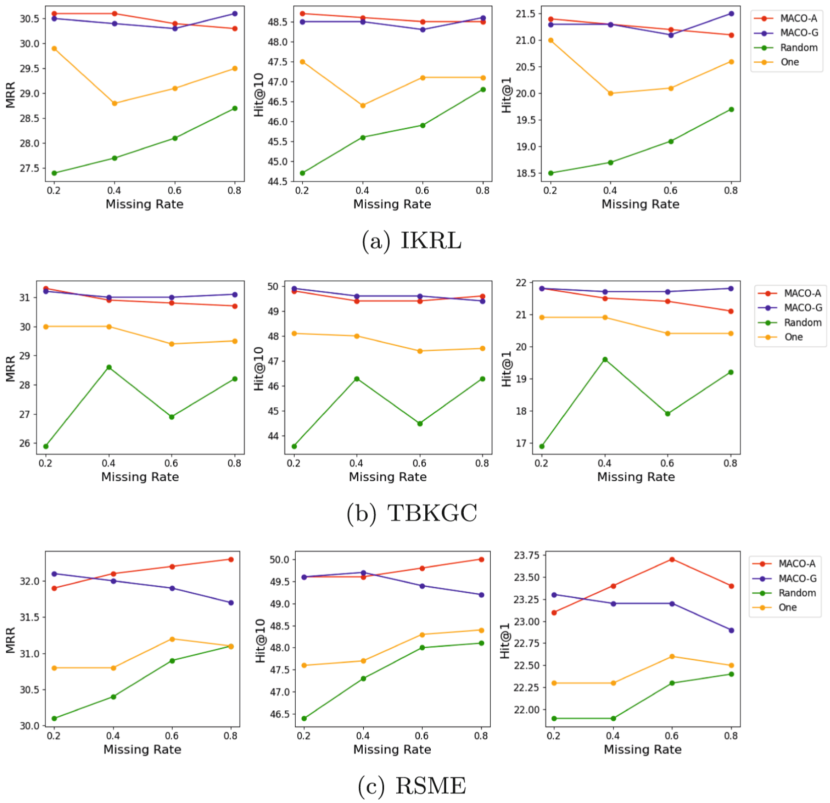
In this section, we will present the detailed experiment settings and the experimental results to demonstrate the effectiveness of MACO. We conduct experiments to answer the following three research questions (RQ) about MACO:

- **RQ1:** Could MACO outperform the baseline methods and achieve state-of-the-art results in KGC task?
- **RQ2:** Is the design of each module in MACO reasonable, and is there a pattern to the selection of hyperparameters?
- **RQ3:** Is there a more intuitive explanation for the performance of MACO?

4.1 Experiment Settings

Datasets. For our experiments, we use FB15K-237 [12] dataset, a public benchmark to conduct our experiments. FB15K-237 has 14541 entities and 237 relations. The train/valid/test set has 272115/17535/20466 triples respectively. The origin FB15K-237 dataset is modality-complete and we construct modality-missing datasets by randomly dropping the visual information of entities with the missing rate (MR) 20%, 40%, 60%, 80% respectively.

Tasks and Evaluation Protocols. We evaluate our method with the link prediction task, which is the main task of KGC. The link prediction task aims to predict the missing entities for the given query $(h, r, ?)$ or $(?, r, t)$. We evaluate our method with mean reciprocal rank (MRR), and Hit@K (K = 1, 3, 10) following [11]. Besides, we follow the filter setting [1] which would remove the candidate triples that have already appeared in the training data to avoid their interference.



(a) IKRL

(b) TBKGC

(c) RSME

Fig. 3. The link prediction results (MRR, Hit@10, Hit@1) compared with baseline methods (Random, One) under different missing rates and different score functions.

Baselines. MACO is designed to complete the missing visual information in the KGs. As few existing works specifically address the modality-missing problem, we have limited choices for baselines. Previous methods often complete the missing modality information by randomly initializing [10] or setting them all to one [14]. We name these two methods random and one for short. Besides, we employ several different score functions (IKRL [17], TBKGC [10], RSME [14]) to demonstrate the generality of MACO.

Parameter Settings. To train MACO, we set the dimension of structural feature and random noise to 768/128, the number of R-GCN layers L to 2, and the training batch size to 128. The dimension of visual feature captured by ViT [3] is 768. The hidden size of the FFN is set to 256 for both $\mathbf{G}$ and $\mathbf{D}$. We train MACO for 500 epoches with learning rate $1e^{-4}$ for both $\mathbf{G}$ and $\mathbf{D}$. The temperature τ is searched in $\{0.5, 1, 2\}$ and α is searched in $\{0.0001, 0.01, 0.1\}$. The number of generated features K is set to 512.

As for the link prediction, we fixed the embedding dimension to 128, the batch size to 1024, and the number of negative samples N to 32. The margin λ is searched in $\{4, 6, 8\}$ and β is set to 2. All experiments are conducted on Nvidia A100 GPUs.

4.2 Main Results (RQ1)

The main results of the link prediction experiments are shown in Fig. 3. From the figures we could observe that MACO could outperform the existing methods on all the evaluation metrics and complete the missing modality with a more semantic-rich representation to achieve better link prediction results under different missing rates with different score functions. Furthermore, we find that the baseline performance is not negatively correlated with the missing rate as expected, but they are significantly lower than MACO, which indicates that vanilla modality completion is not stable for the utilization of modal information.

Besides, the two strategies (All-Gen and Gen) of MACO exhibit similar performance. They are model-specific as their performance varies across different score functions. For example, All-Gen performs better in RSME, while Gen performs better in TBKGC generally. Compared to the baseline, the experimental results are more stable under different missing rates, which reflects that MACO could model the distribution of visual information in the graph structure well and generate robust visual representations for entities.

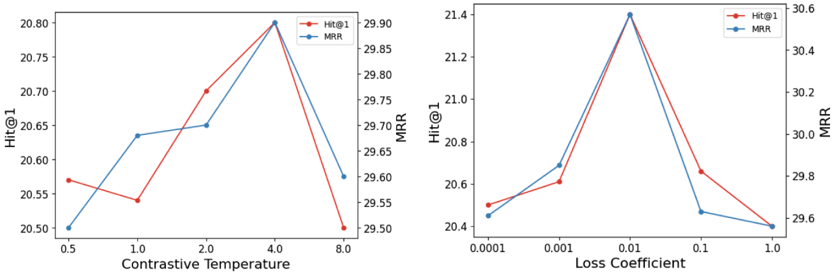
4.3 Further Analysis (RQ2)

Ablation Study. To answer **RQ2**, we conduct ablation study and parameter analysis on MACO to demonstrate the effectiveness of each module and hyper-parameters in MACO. In ablation study, we mainly focus on three aspects: (1) the modality-adversarial generator (w/o MA), (2) the R-GCN structural encoder (w/o SE), (3) the contrastive loss function (w/o CL). We remove the mentioned modules respectively and conduct link prediction experiments to explore the quality of the generatedvisual features. Table 1 displays the detailed settings and ablation study results, which show that removing any of the modules causes a degradation in results on both score functions. The ablation study indicates that extracting the graph structural features by graph encoder and treating them as the condition of the generator while applying contrastive loss on the features could improve the quality of the generated visual features.

Parameter Analysis. We further evaluate the influence of the hyper-parameters of MACO including the temperature τ and the contrastive loss coefficient α, which are newly introduced in MACO. Figure 4 reveals that the two hyper-parameters significantly affect the model's performance. Additionally, they show a similar impact on the model performance, with an initial improvement observed as the hyperparameters increase, which is later followed by a performance decrease. Empirically speaking, the optimal choice of tau and α is near 4.0 and 0.01 respectively.

Table 1. The ablation study results. We set the missing rate as 40%. For the model w/o MA, we replace **G** with an unconditional generator. For the model w/o SE, we replace the R-GCN encoder with a vanilla embedding layer. For the model w/o CL, we remove the contrastive loss on the training objective.

Model	IKRL				RSME			
	MRR	Hit@10	Hit@3	Hit@1	MRR	Hit@10	Hit@3	Hit@1
MACO	30.6	48.6	34.1	21.3	32.1	49.6	35.1	23.4
w/o MA	29.6	47.5	32.8	20.4	31.4	48.5	34.4	22.8
w/o SE	29.6	47.6	32.8	20.6	31.3	48.2	34.3	22.7
w/o CL	29.7	47.7	32.9	20.7	31.3	48.4	34.3	22.8



(a) Constrative Temperature

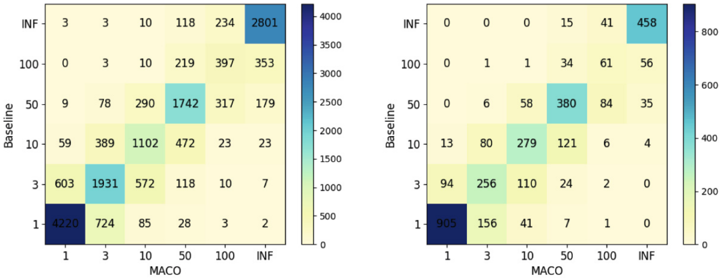
(b) Loss Coefficient

Fig. 4. Parameter analysis results of MACO. The missing rate of dataset is 60% and IKRL [17] score function is employed for the parameter analysis.

4.4 Case Study (RQ3)

To illustrate the effectiveness of MACO and answer **RQ3**, we further conduct a case study. We divide the triples in the test set into two categories based on whether or not there is a modality-missing entity in the triple. Besides, we draw the heat maps of the link prediction results between the MACO-enhanced model and the baseline model. The heat maps are shown in Fig. 5, where the lower half of the diagonal indicates the triples where the MACO method outperformed the baseline model, and the upper half indicates the opposite.

We find though some triples get worse rankings, more modality-missing triples achieve better ranks with the help of MACO, which reflects that MACO could complete the missing visual information with semantic-rich visual representations. For example, given the test triple *(Michael Gough, /film/actor, Batman)*, the tail entity *Batman* is modality-missing. Typically, the visual information of the film might be a poster which is important information to match the actors, which is similar to the case mentioned in Fig. 1. Thus, the modality-missing situation makes the predicted rank of the baseline model 60. However, the model enhanced by MACO predicts the correct tail entity with rank 1. Such a simple case intuitively demonstrates the effectiveness of MACO. Besides, we



(a) Modality-missing triples (b) Modality-complete triples

Fig. 5. Heat map visualization for the link prediction results. In each heat map, the x-axis/y-axis represents the link prediction results enhanced by MACO and random completion respectively. We divided the linked prediction results into six intervals.

could conclude that All-Gen could also benefit those modality-complete triples by generating high-quality visual representations and replacing the original ones.

5 Conclusion

In this paper, we mainly discuss the modality-missing problem in the existing MMKGC methods. We argue that vanilla approaches like random initialization would introduce noise into the MMKGC model, leading to bad performance. We propose MACO, a modality adversarial and contrastive framework that generates visual modal features of entities conditioned on structural information to preserve the correspondence between the structure and visual information. This approach completes modality-missing entities with semantic-rich visual representations. We conduct experiments on public benchmarks to demonstrate the effectiveness of MACO. In the future, we plan to collaborative the collaborative design of missing-modality completion and knowledge graph completion.

Acknowledgement. This work is funded by Zhejiang Provincial Natural Science Foundation of China (No. LQ23F020017), Yongjiang Talent Introduction Programme (2022A-238-G), and NSFC91846204/U19B2027.

References

1. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NeurIPS (2013)
2. Cao, Z., Xu, Q., Yang, Z., He, Y., Cao, X., Huang, Q.: OTKGE: multi-modal knowledge graph embeddings via optimal transport. In: Proceedings of NeurIPS (2022)
3. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

4. Du, C., et al.: Semi-supervised deep generative modelling of incomplete multi-modality emotional data. In: 2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, 22–26 October 2018 (2018)
5. Goodfellow, I.J., et al.: Generative adversarial nets. In: Proceedings of NeurIPS (2014)
6. Jing, M., Li, J., Zhu, L., Lu, K., Yang, Y., Huang, Z.: Incomplete cross-modal retrieval with dual-aligned variational autoencoders. In: Proceedings of ACM MM (2020)
7. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of ICLR (2014)
8. Lee, K.S., Tran, N.T., Cheung, N.M.: Infomax-GAN: improved adversarial image generation via information maximization and contrastive learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2021)
9. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
10. Sergieh, H.M., Botschen, T., Gurevych, I., Roth, S.: A multimodal translation-based approach for knowledge graph representation learning. In: Proceedings of AACL (2018)
11. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: Proceedings of ICLR (2019)
12. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of EMNLP (2015)
13. Trouillon, T., Dance, C.R., Gaussier, É., Welbl, J., Riedel, S., Bouchard, G.: Knowledge graph completion via complex tensor factorization. J. Mach. Learn. Res. (2017)
14. Wang, M., Wang, S., Yang, H., Zhang, Z., Chen, X., Qi, G.: Is visual context really helpful for knowledge graph? A representation learning perspective. In: MM 2021: ACM Multimedia Conference, Virtual Event, China, 20–24 October 2021 (2021)
15. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**, 2724–2743 (2017)
16. Wang, Z., Li, L., Li, Q., Zeng, D.: Multimodal data enhanced representation learning for knowledge graphs. In: Proceedings of IJCNN (2019)
17. Xie, R., Liu, Z., Luan, H., Sun, M.: Image-embodied knowledge representation learning. In: Proceedings of IJCAI (2017)
18. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
19. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of ICLR (2015)
20. Zhang, H., Koh, J.Y., Baldridge, J., Lee, H., Yang, Y.: Cross-modal contrastive learning for text-to-image generation. In: Proceedings of CVPR (2021)
21. Zhao, Y., et al.: MoSE: modality split and ensemble for multimodal knowledge graph completion. In: Proceedings of EMNLP (2022)

A Joint Entity and Relation Extraction Approach Using Dilated Convolution and Context Fusion

Wenjun Kong, Yamei Xia(✉), Wenbin Yao, and Tianbo Lu

School of Computer Science, (National Pilot Software Engineering School),
Beijing University of Posts and Telecommunications, Beijing, China
{kongwenjun,ymxia,yaowenbin,lutb}@bupt.edu.cn

Abstract. In recent years, researchers have shown increasing interest in joint entity and relation extraction. However, existing approaches overlook the interaction between words at different distances and the significance of contextual information between entities. We believe that the correlation strength of word pairs should be considered, and it is necessary to integrate contextual information into entities to learn better entity-level representations. In this paper, we treat named entity recognition as a multi-class classification of word pairs. We employ self-attention mechanism and design both local and multi-grained dilated convolution layers to capture spatial correlations between words. In the relation extraction module, we leverage attention from the self-attention layer to fuse localized context information into entity-pair to produce context-enhanced entity-level representations. In addition, we integrate named entity recognition and relation extraction through a multi-task learning framework, effectively leveraging the interaction between two subtasks. To validate the performance of our model, we conducted extensive experiments on joint entity and relation extraction benchmark datasets CoNLL04, ADE and SciERC. The experimental results indicate that our proposed model can achieve significant improvements over existing methods on these datasets.

Keywords: Joint entity and relation extraction · Self-attention · Dilated convolution · Context fusion

1 Introduction

Named entity recognition (NER) and relation extraction (RE) are the core tasks of natural language processing and information extraction, which can extract the type of entities and the semantic relationship between entity pairs from text. They are important research objects in the fields of information retrieval, question answering [1] and knowledge graph [2].

Traditionally, this problem is decomposed into two pipeline subtasks. However, the pipeline method often has problems such as error propagation and poor interactivity between two tasks, resulting in performance bottlenecks.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 135–146, 2023.
https://doi.org/10.1007/978-3-031-44693-1_11

Recently, some joint neural models [3–6] have been proposed to detect entities and their relations, which alleviate the above problems to a certain extent and achieve better performance. However, these existing models fail to fully exploit spatial correlations between words, despite the rich and intricate semantic relationships that exist among them. Besides, due to pairwise enumeration of entities in the task of relation extraction, an entity will appear in different entity pairs. Existing works do not construct specific contextualized entity-level representation for an entity appears in different entity pairs. Instead, they use the same entity representation, neglecting the localized context information, which often leads to suboptimal performance.

To address above problems, we propose a joint entity and relation extraction neural model that combines self-attention mechanism [7], both local and multi-grained dilated convolution and localized context fusion. In the NER module, we employ the self-attention mechanism to produce a word-pair correlation matrix, where each element represents a span. In order to capture the spatial semantics between words, we introduce relative distance embedding and use both local convolution and multi-grained dilated convolution to refine the features of words at varying distances. Following the entity recognition step, we leverage the attention obtained from the self-attention layer to incorporate contextual information into entity representations, thereby enabling the model to take account into potential clues from the localized context to predict relations. Finally, we integrate named entity recognition and relation extraction through a multi-task learning framework, effectively utilizing the interaction between these two tasks.

In summary, our contributions are as follows:

- We propose a joint entity and relation extraction model that applies self-attention mechanism and both local and multi-grained dilated convolutional neural network, which can effectively capture the semantic relationship between words.
- We propose a context-enhanced entity representation fusion method, which utilizes the attention weights of entity pairs obtained from the self-attention layer to fuse specific localized context into entity representations.
- Compared with existing models, our approach has shown a significant F1 score improvement, and the effectiveness of our approach has been demonstrated in ablation experiments.

2 Related Work

Traditionally, named entity recognition [8, 9] and relation extraction [10] have been implemented as two independent pipeline tasks. The pipeline approach is relatively simpler in modeling, but there are obviously a series of problems: (1) error propagation: the performance of NER models directly affects the results of relation extraction. (2) poor interactivity: the interaction between these two subtasks is not considered, resulting in performance bottlenecks.

Recently, some joint neural models have been proposed for entity and relation extraction, which alleviate the error propagation problem to a certain extent, and explore the interaction between NER and RE. Most of joint models can be divided into three classes: sequence labeling, table filling and span-based.

Sequence Labeling. Bekoulis et al. [3] viewed entity recognition as a sequence labeling task, and relation extraction as a multi-head selection problem, identifying potential multiple relations for each entity. Bekoulis et al. [11] use adversarial training for the tasks of entity recognition and relation extraction, improving the robustness of neural model. Miwa and Bansal [12] use BiLSTM [13] to extract the features of entities and tree-LSTM [14] to model the relations between the entities.

Table Filling. Miwa and Sasaki [15] treated joint entity and relation extraction as a table filling problem. The diagonal lines of the table serve as labels for entities, and the remaining parts serve as labels for relations. Similarly, Zhang et al. [6] propose a global optimization method to fill the table. Wang and Lu [4] design two distinct encoders to capture table and sequence features respectively. Yan et al. [16] introduced a partition filter network for modeling the two-way interaction between subtasks and classifying entities and relations using the table-filling approach. Wang et al. [17] unified the label space for two subtasks by representing entities and relations with rectangles in a table format.

Span-Based. Dixit and Al [18] use span representations to extract relations between entities. Luan et al. [19] treat entity extraction as a span classification problem and construct the dynamic span graph to refine the span representations. Ebert and Ulges [5] enumerate all the possible spans in a sentence and apply the max-pooling method to extract span features.

However, most models mentioned above fall short in capturing the spatial correlations between words and the localized context information between entities. It should be considered that the spatial associations between words can convey rich entity information. Additionally, we believe that incorporating localized context between entities is beneficial for learning better entity-level representations to improve relation extraction.

3 Methodology

In this section, we first formally define the problem of joint entity recognition and relation extraction in Sect. 3.1 and then detail our approach in subsequent Sects. 3.2–3.5. Overall framework of our model is shown in Fig. 1.

3.1 Problem Definition

Joint Entity and Relation Extraction. The input of the problem is a sentence X consisting of n tokens $\mathrm{X} = \{x_1, x_2, \ldots, x_n\}$. Let E denote a set of pre-defined entity types, and T denote a set of pre-defined relation types. Let $\mathrm{S} = \{s_1, s_2, \ldots, s_m\}$ be all the possible spans in sentence X. The named entity recognition task is, for each span $s_i \in \mathrm{S}$, to predict an entity type $\mathrm{e} \in \mathrm{E}$. The relation extraction task is, for spans of each entity pair $s_i, s_j \in \mathrm{S}$, to predict a relation type $t \in \mathrm{T}$. The output of joint entity and relation extraction is a set of triplets $\mathrm{Y} = \left\{\left((s_i, e_i), \left(s_j, e_j\right), t\right) : s_i, s_j \in \mathrm{S}, e_i, e_j \in \mathrm{E}, t \in \mathrm{T}\right\}$.

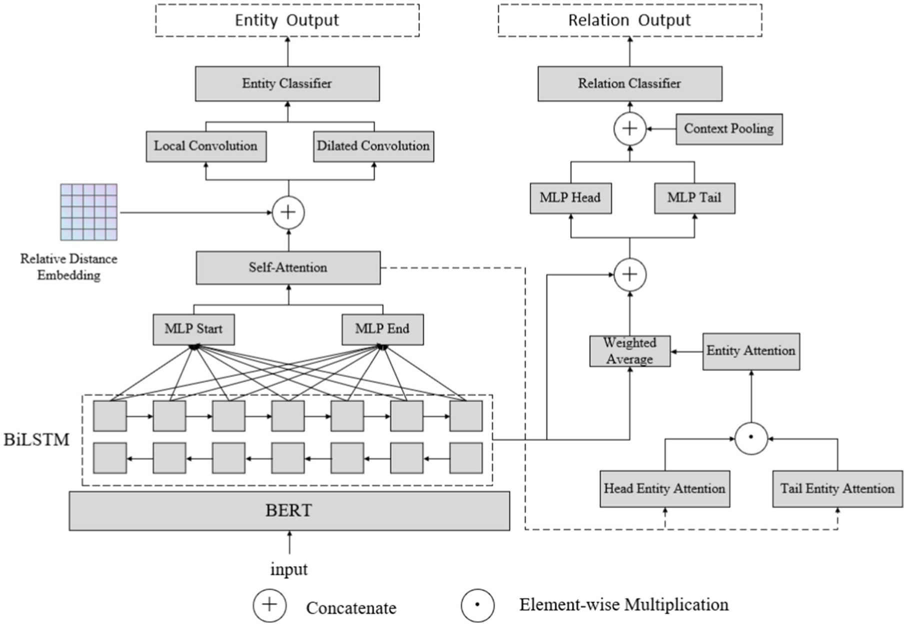


Fig. 1. The framework of our model consists of two task modules: Entity Recognition Module and Relation Extraction Module.

3.2 Representation Learning

We use BERT [20] as our pre-trained model. Given an input sentence, BERT tokenizer will convert each token into word pieces. However, it is not desirable for the model to predict specific word pieces as entity boundaries, which cause hallucinations during the training or inference. To eliminate this negative impact, we employ max-pooling to produce token representations based on the word piece representations. Then, we further use a bi-directional LSTM to refine token representations:

$$\mathrm{X}' = \mathrm{MaxPooling}(\mathrm{BERT}(\mathrm{X})) \tag{1}$$

$$\mathrm{H} = \mathrm{BiLSTM}(\mathrm{X}') \tag{2}$$

3.3 Entity Recognition Module

To represent the start and end of the entity span, we first apply two multi-layer perception (MLP) to map each token representation into the start and end of span vector representation space separately. By means of separate representations for the beginning and end of the span, the model could recognize the boundary of the entity.

$$h_{s_i} = \mathrm{MLP}(h_i) \tag{3}$$

$$h_{e_i} = \mathrm{MLP}(h_i) \tag{4}$$

where h_i means the hidden representation of i-th token.

In order to learn the correlation between words, we utilized self-attention mechanism convert a sequence into a word-pair correlation matrix, which is calculated as follow:

$$\mathrm{A}_{ij} = W\left(\tanh\left(h_{s_i} + h_{e_j}\right)\right) + b \tag{5}$$

where $W \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^{d}$ are learnable parameters, $\mathrm{A} \in \mathbb{R}^{n\times n\times d}$.

We regard word-pair correlation matrix A as an image. Inspired by Li et al. [21], we propose to employ local convolution module and dilated convolution module, both of which are parallel. The local convolution layer consists several 3 × 3 convolutional layers stacked together to extract local features of word-pairs. The dilated convolution module overcomes the limitation of local convolution by incorporating multi-grained dilated convolutional layers, which increase the receptive field size, allowing the model to capture features from words at varying distances. The dilated convolutional layers with different dilated rates express features at various levels. To ensure coverage of features across all levels, we concatenate the hidden states of each dilated convolutional layer through skip connections, as shown in Fig. 2. By combining local convolution and dilated convolution, our model can effectively capture fine-grained spatial relationships between words.

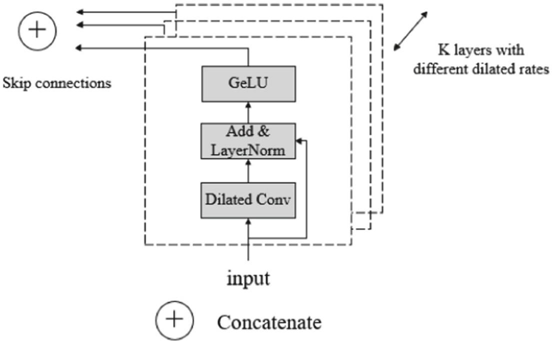


Fig. 2. The dilated convolution module consists of K dilated convolutional layers, each with different dilated rates. We utilize skip connections to concatenate the outputs of these layers.

Additionally, in order to learn better spatial semantics between words and prevent hallucination of entity boundaries, we introduce a relative distance embedding matrix $\mathrm{D} \in \mathbb{R}^{n\times n\times d'}$, which is concatenated as the input of the convolutional layer.

$$\begin{aligned}
\mathrm{A}' &= \mathrm{A} \oplus \mathrm{D} \\
\mathrm{Q}_{loc} &= \mathrm{GeLU}\left(\mathrm{LayerNorm}\left(\mathrm{Conv2d}\left(\mathrm{A}'\right) + \mathrm{A}'\right)\right) \\
\mathrm{Q}_{dil}^{j} &= \mathrm{GeLU}\left(\mathrm{LayerNorm}\left(\mathrm{DilatedConv2d}\left(\mathrm{Q}_{dil}^{j-1}\right) + \mathrm{Q}_{dil}^{j-1}\right)\right)
\end{aligned} \tag{6}$$

where $\oplus$ refers to concatenate, $\mathrm{GeLU}(\cdot)$ denotes Gaussian error linear units [22] activation function, $\mathrm{LayerNorm}(\cdot)$ denotes layer normalization [23], Q_{loc} denotes the output of the local convolutional layer, Q_{dil}^{j} denotes the output of the j-th dilated convolutional layer with specific dilated rate, and Q_{dil}^{0} is equal to A′.

Finally, we concatenate the outputs of both the local and dilated convolutional layers, and then feed them into a MLP to calculate the scores for all spans.

$$\mathrm{W} = \mathrm{MLP}\Big(\mathrm{Q}_{loc} \oplus \mathrm{Q}_{dil}^{1} \oplus \mathrm{Q}_{dil}^{2} \oplus \ldots \oplus \mathrm{Q}_{dil}^{K}\Big) \tag{7}$$

3.4 Relation Extraction Module

Contextual information is crucial for relation classification task. Consequently, we believe that integrating localized context information into entity-pair representations would be beneficial. This integration allows us to construct specific contextualized entity-level representations for an entity that appears in different entity pairs.

Specifically, with the word-pair correlation matrix A obtained from Eq. 5, we calculate the average attention score for each word within an entity, and then consider it as the entity-level attention. Inspired by Zhou et al. [24], we pairwise enumerate entities and multiply their entity-level attention to weight the token representations as the localized context representation for the entity pair. As a result, tokens that are more significant for the entity pair will have higher weights.

$$\begin{aligned}
\hat{A}_{e_k} &= \frac{1}{end_k - start_k + 1} \sum_{i=start_k}^{end_k} \alpha_i \\
\hat{A}^{(e_i,e_j)} &= \hat{A}_{e_i} \circ \hat{A}_{e_j} \\
\mathrm{V}^{(e_i,e_j)} &= \mathrm{H}^{\mathrm{T}} \hat{A}^{(e_i,e_j)}
\end{aligned} \tag{8}$$

where $\alpha = \mathrm{Softmax}(\mathrm{FFN}(\mathrm{A}))$, $start_k$, end_k represent the start index and end index of the k-th entity span, $\hat{A}_{e_k} \in \mathbb{R}^{n \times 1}$, $\circ$ denotes element-wise multiplication, H is token representations calculated by Eq. 2.

We employ max-pooling to produce span representations. Next, we combine the entity span representation with the localized contextual representation, and use two MLPs to separately learn the representations of the head entity and the tail entity.

$$\mathrm{U}_{head} = \mathrm{MLP}\Big(\mathrm{V}^{(e_i,e_j)} \oplus h(e_i)\Big) \tag{9}$$

$$\mathrm{U}_{tail} = \mathrm{MLP}\Big(\mathrm{V}^{(e_i,e_j)} \oplus h\big(e_j\big)\Big) \tag{10}$$

where $h(e_i)$ represents the representation of the entity span e_i.

Besides incorporating context into the entity-level representations, we propose to employ max-pooling on the token representations between the head entity and the tail entity to obtain localized contextual representation $c\big(e_i, e_j\big)$. We concatenate head entity representation U_{head} and tail entity representation U_{tail}, calculated by Eq. 9 and Eq. 10, with the localized contextual representation $c\big(e_i, e_j\big)$, and use this concatenated representation as the input for the final classification layer:

$$\mathrm{Z} = \mathrm{MLP}\big(\mathrm{U}_{head} \oplus \mathrm{U}_{tail} \oplus c\big(e_i, e_j\big)\big) \tag{11}$$

Utilizing this module, entities can be represented by fusing information from their specific context, allowing the model to leverage potential clues from the localized context for predicting relations.

3.5 Joint Learning

We treat named entity recognition as a multi-class classification problem. According to the entity classification score calculated by Eq. 7, we use softmax cross-entropy as the loss of entity recognition module:

$$\mathcal{L}_{NER} = -\sum_{0 \le i \le j < n} \sum_{t \in E} \hat{e}_{ij}^{t} \log\left(\mathrm{W}_{ij}^{t}\right) \tag{12}$$

where $\hat{e}_{ij}^{t}$ denotes the gold distribution of span from position i to j with entity type t.

Multiple relations may exist between two entities. In order to extract these multiple relations, we regard relation extraction as a multi-label classification problem, and use binary cross-entropy as the loss of the relation extraction module:

$$\mathcal{L}_{RE} = -\sum_{i=1}^{M} \sum_{t \in T} \hat{r}_{i}^{t} \log\left(\mathrm{Z}_{i}^{t}\right) + \left(1 - \hat{r}_{i}^{t}\right) \log\left(1 - \mathrm{Z}_{i}^{t}\right) \tag{13}$$

where $\hat{r}_{i}^{t}$ represents the gold distribution of i-th entity pair with relation t.

The model is trained by minimizing the weighted sum of NER loss and RE loss:

$$\mathcal{L} = \alpha \mathcal{L}_{NER} + \beta \mathcal{L}_{RE} \tag{14}$$

where α and β are weight hyperparameters.

4 Experiments

In this section, we first introduce the setup of experiments in Sect. 4.1, and then describe our implementation details in Sect. 4.2. Finally, we analyze our main experimental results in Sect. 4.3 and ablation experiments results in Sect. 4.4.

4.1 Setup

Datasets. We evaluate our approach on three popular joint named entity recognition and relation extraction datasets: CoNLL04 [25], ADE [26] and SciERC [27]. The CoNLL04 dataset consists of annotated sentences from news articles, featuring four entity types (*Location, Organization, Person, Other*) and five relation types (*Work-For, Kill, Organization-Based-In, Live-In, Located-In*). The ADE comprises data from numerous reports on drug-related safety issues, with one relation type (*Adverse-Effect*) and two entity types (*Adverse-Effect* and *Drug*). The SciERC dataset is designed to define scientific entities and relations for constructing scientific knowledge graphs. We follow previous work [5] and use the same splits for all the datasets.

Evaluation Metrics. We evaluate our model on both NER and RE. An entity is regarded as correct only if its predicted span and entity type are both correct. A relation between two entities is considered correct if its relation type as well as its related two entities are both match the ground truth. Following the previous work, we measure the precision, recall, micro-F1 and macro-F1 score for entities and relations. For ADE, the above metrics are averaged over the 10-folds cross-validation.

4.2 Implementation Details

For a sentence of length n, there are $\frac{n\times(n+1)}{2}$ spans. Typically, only a few spans are labeled as entities, while "non-entity" spans accounted for the vast majority, which will lead to an imbalance in the distribution of positive and negative samples. To relieve this issue and facilitate model convergence, we employ negative sampling during the training phase. Specifically, for each sentence, we randomly sample 40% of the total number of spans without an entity label as negative example. Additionally, we incorporate all labeled entities as inputs to the RE module. During the inference, following the setup of Yu et al. [28], we rank all the spans that have a category other than "non-entity" by their category scores and apply post-processing constraints to entities with clash boundaries. Given a candidate entity pair $\left(s_i, s_j\right)$, the relation between two entities is predicted as t_k only if $P\left(t_k|s_i, s_j\right)$. is greater than a fixed relation classification threshold δ. We set threshold $\delta = 0.4$.

For CoNLL04 and ADE, we use bert-base-cased [20] as the pretrained model. For SciERC, we use scibert-scivocab-cased [29], which has shown to be more effective than BERT in this specific domain due to its in-domain pre-training.

4.3 Main Results

As is shown in Table 1, our proposed model achieves strong performance compared to existing models. In comparison with SpERT [5], our model has achieved an absolute entity micro-F1 improvement of +1.40%, +0.83%, and relation micro-F1 improvement of +3.16%, +2.60% on CoNLL04 and ADE respectively. Remarkably, our model based on bert-base-cased has achieved competitive performance, even outperforming Table-Sequence [4] based on BERT variant – albert-xxlarge-v1 [30], with its parameters approximately twice the size of ours. For SciERC, our model also achieves superior performance, with entity and relation micro-F1 surpassing the existing strong baselines by 2.14% and 2.63% respectively. These improvements demonstrate our model can effectively learn the spatial semantics between words as well as localized context information between entities.

4.4 Ablation Study

We choose to remove certain parts of our model, Table 2 shows ablation study on CoNLL04. In the case of removing relative distance embedding, we find that entity F1 decreases by 0.56%. Relative distance embedding can provide useful spatial information for convolutional layers and improve the accuracy of entity boundary recognition. Although relative distance embedding is designed for NER tasks, without relative

Table 1. The experiment results compared with existing methods ($\dagger$ = micro-average, $\ddagger$ = macro-average, * = not stated, P = precision, R = recall, PLM = pretrained language model, B = bert-base, ALB = albert-xxlarge-v1, SciB = scibert-base). To ensure fair comparison, we report the results of PURE in a single-sentence setting.

Dataset	Model	PLM	Entity			Relation		
			P	R	F1	P	R	F1
CoNLL04	Global Optimization [6]†	–	–	–	85.60	–	–	67.80
	Multi-turn QA [31]†	B	89.00	86.60	87.80	69.20	68.20	68.90
	Hierarchical Attention [32]*	–	–	–	86.51	–	–	62.32
	SpERT [5]†	B	88.25	89.64	88.94	73.04	70.00	71.47
	SpERT [5]‡		85.78	86.84	86.25	74.75	71.52	72.87
	Table-Sequence [4]†	ALB	–	–	90.10	–	–	73.60
	Ours†	B	**90.13**	**90.55**	**90.34**	**78.53**	**71.09**	**74.63**
	Ours‡		**87.46**	**87.18**	**87.32**	**79.10**	**72.88**	**75.86**
ADE	BiLSTM + SDP [33]*	–	82.70	86.70	84.60	67.50	75.80	71.40
	SpERT [5]†	B	**88.69**	89.20	88.95	77.77	79.96	78.84
	SpERT [5]‡		88.99	89.59	89.28			
	Table-Sequence [4]‡	ALB	–	–	89.70	–	–	80.10
	PFN [16]‡	B	–	–	89.60	–	–	80.00
	Ours†	B	88.66	**90.93**	**89.78**	**79.91**	**83.03**	**81.44**
	Ours‡		**89.15**	**91.93**	**90.52**			
SciERC	UNIRE [17]†	SciB	65.80	71.10	68.40	37.30	36.60	36.90
	PURE [34]†	SciB	–	–	66.60	–	–	35.60
	PFN [16]†	SciB	–	–	66.80	40.60	36.50	38.40
	Ours†	SciB	**69.70**	**71.39**	**70.54**	**41.09**	**40.97**	**41.03**
	Ours‡		**69.50**	**71.27**	**70.37**	**40.15**	**39.72**	**39.93**

distance embedding, the relation F1 also slightly declines. This may result from the interaction between the two tasks in the joint model, indicating that better entity recognition contributes to the learning of relation extraction module. In the case of removing multi-grained dilated convolution module, and only the local convolution layer is used instead, the entity F1 and the relation F1 decrease by 0.83% and 0.72% respectively. We can conclude that multi-grained dilated convolution can compensate for the shortcomings of local convolution, and effectively capture the spatial semantics between words with different distances. By combining local convolution and multi-grained dilated convolution, the model can produce better spatial refinement representations. Furthermore, we conducted experiments without context fusion and observed that the performance of entity extraction was minimally affected. However, there was a decrease of 0.97% in relation F1, suggesting that the context fusion approach plays a crucial role in learning better entity-level representations, consequently enhancing the performance of relation extraction.

Table 2. Ablation study on CoNLL04.

Setting	Entity F1	Δ	Relation F1	Δ
Default	90.34	/	74.63	/
–Relative distance embedding	89.78	0.56	74.28	0.35
–Dilated convolution layers	89.51	0.83	73.91	0.72
–Context fusion	90.21	0.13	73.66	0.97

5 Conclusion

In this paper, we propose a joint entity and relation extraction neural model, which combines self-attention mechanism, both local and multi-grained dilated convolution and localized context fusion. Our NER module incorporates local and multi-grained dilated convolutions to capture spatial semantics between words. In the RE module, a context fusion approach is employed to enhance entity-level representations by incorporating contextual information. Finally, we integrate these two subtasks into a multi-task learning framework, fully utilizing the interaction between NER and RE tasks. The experimental results indicate that our proposed model can achieve significant improvements over existing methods and the ablation experiments demonstrate the effectiveness of our approach. For the future work, we hope to extend our model to the domain of document-level joint entity and relation extraction.

Acknowledgement. We thank the anonymous reviewers for their helpful comments and feedback. This work is supported by the National Natural Science Foundation of China (62162060).

References

1. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. In: Proceedings of EMNLP, pp. 615–620 (2014)
2. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Netw. Learn. Syst. **33**, 494–514 (2022)

3. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst. Appl. **114**, 34–45 (2018)
4. Wang, J., Lu, W.: Two are better than one: joint entity and relation extraction with table-sequence encoders. In: Proceedings of EMNLP, pp. 1706–1721 (2020)
5. Eberts, M., Ulges, A.: Span-based joint entity and relation extraction with transformer pre-training. In: ECAI, pp. 2006–2013 (2019)
6. Zhang, M., Zhang, Y., Fu, G.: End-to-end neural relation extraction with global optimization. In: Proceedings of EMNLP, pp. 1730–1740 (2017)
7. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS, pp. 5998–6008 (2017)
8. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
9. Souza, F., Nogueira, R., Lotufo, R.: Portuguese named entity recognition using BERT-CRF. arXiv preprint arXiv:1909.10649 (2019)
10. Wang, H., et al.: Extracting multiple-relations in one-pass with pre-trained transformers. In: Proceedings of ACL, pp. 1371–1377 (2019)
11. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Adversarial training for multi-context joint entity and relation extraction. In: Proceedings of EMNLP, pp. 2830–2836 (2018)
12. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of ACL (Volume 1: Long Papers), pp. 1105–1116 (2016)
13. Graves, A., Jaitly, N., Mohamed, A.: Hybrid speech recognition with deep bidirectional LSTM. In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 273–278 (2013)
14. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of ACL-IJCNLP (Volume 1: Long Papers), pp. 1556–1566 (2015)
15. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of EMNLP, pp. 1858–1869 (2014)
16. Yan, Z., Zhang, C., Fu, J., Zhang, Q., Wei, Z.: A partition filter network for joint entity and relation extraction. In: Proceedings of EMNLP, pp. 185–197 (2021)
17. Wang, Y., Sun, C., Wu, Y., Zhou, H., Li, L., Yan, J.: UniRE: a unified label space for entity relation extraction. In: Proceedings of ACL-IJCNLP (Volume 1: Long Papers), pp. 220–231 (2021)
18. Dixit, K., Al-Onaizan, Y.: Span-level model for relation extraction. In: Proceedings of ACL, pp. 5308–5314 (2019)
19. Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., Hajishirzi, H.: A general framework for information extraction using dynamic span graphs. In: Proceedings of NAACL-HLT, Volume 1 (Long and Short Papers), pp. 3036–3046 (2019)
20. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
21. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: Proceedings of AAAI, pp. 10965–10973 (2022)
22. Hendrycks, D., Gimpel, K.: Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415 (2016)
23. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
24. Zhou, W., Huang, K., Ma, T., Huang, J.: Document-level relation extraction with adaptive thresholding and localized context pooling. In: Proceedings of AAAI, pp. 14612–14620 (2021)

25. Roth, D., Yih, W.T.: A linear programming formulation for global inference in natural language tasks. In: Proceedings of HLT-NAACL, pp. 1–8 (2004)
26. Gurulingappa, H., Rajput, A.M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L.: Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. J. Biomed. Inform. **45**(5), 885–892 (2012)
27. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Proceedings of EMNLP, pp. 3219–3232 (2018)
28. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of ACL, pp. 6470–6476 (2020)
29. Beltagy, I., Lo, K., Cohan, A.: SciBERT: a pretrained language model for scientific text. In: Proceedings of EMNLP-IJCNLP, pp. 3615–3620 (2019)
30. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
31. Li, X., et al.: Entity-relation extraction as multi-turn question answering. In: Proceedings of ACL, pp. 1340–1350 (2019)
32. Chi, R., Wu, B., Hu, L., Zhang, Y.: Enhancing joint entity and relation extraction with language modeling and hierarchical attention. In: APWeb-WAIM, pp. 314–328 (2019)
33. Li, F., Zhang, M., Fu, G., Ji, D.: A neural joint model for entity and relation extraction from biomedical text. BMC Bioinform. **18**(1), 1–11 (2017)
34. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Proceedings of NAACL-HLT, pp. 50–61 (2021)

Interest Aware Dual-Channel Graph Contrastive Learning for Session-Based Recommendation

Sichen Liu, Shumin Shi(✉), and Dongyang Liu

School of Computer Science and Technology, Beijing Institute of Technology,
Beijing 100081, China
{lsc1999,bjssm,ldy1998}@bit.edu.cn

Abstract. The key issue of session-based recommendation (SBR) is how to efficiently predict the next interaction item based on the item sequence of anonymous users. In order to mine the complex multivariate relationship between items and sessions, we propose a novel model for session-based recommendation named Interest aware Dual-channel Graph Contrastive learning (IDGC). By generating hypergraph and global graph, we focus on item relationships in different aspects, and we create the dual-channel interest-item embedding learning module to dig the higher-order relationships between items and users' interests. To deal with the problem of long-distance information transmission between non-adjacent items, we set the interest node in each session for interest awareness and base on the contrastive learning strategy to enrich the information of the two graphs. At the same time, we exploit position information and time interval information to enhance the session representation. Extensive experiments show that IDGC has significant performance improvement on all evaluation metrics on three benchmark datasets.

Keywords: Session-based recommendation · Interest aware · Dual-channel graph neural network · Contrastive learning

1 Introduction

In the Internet era, massive information often leads to a high degree of information redundancy. Recommendation systems can combine user interests, and make recommendations to the users. The traditional recommendation system mainly uses the known information of the user. However, in many real-world scenarios, the users' information is unknown due to a series of reasons such as privacy protection. So, the session-based recommendation system is gradually emerging. Session-based recommender systems use the behavior sequence of anonymous users over a period of time to predict the next item. They can learn the user's preferences according to the items that users interact with in the session, and provide accurate personalized recommendations for users.

In the early work, the approach of SBR(session-based recommendation) is proposed based on machine learning [12]. With the development of deep learning, Recurrent Neural Networks(RNNs) begin to be applied to SBR [5,13], but

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 147–158, 2023.
https://doi.org/10.1007/978-3-031-44693-1_12

the transition of session data is not strictly sequential, so Graph Neural Networks(GNN) are widely used in SBR [14,15].

However, in the real world, the appearance of an item is often the result of the joint action of a series of previous items. The relationship between items is not a simple binary relationship, but a more complex many-to-many relationship. An item can have multiple associations with multiple different items, and each other item can also have multiple associations with multiple different items. This many-to-many relationship involves the interaction, similarity, and interactivity between items. How to model the complex higher-order relationships between items more accurately is a challenge in the field of session-based recommendation.

Another challenge of session-based recommendation is the sparseness of data. In the application scenario of session recommendation, only a small amount of existing data can be used to predict the users' next behavior. Some recent session-based recommendation works use contrastive learning to enhance information between different views [16,17]. However, how to construct appropriate positive and negative samples and select comparison methods in session-based recommendation is still a problem to be studied.

In order to solve the above problems, we propose a novel model named Interest aware Dual-channel Graph Contrastive learning for session-based recommendation (IDGC). We generate two different graphs to focus on different aspects of information. We construct the interest item interaction hypergraph, and use the hypergraph structure [8,17] to mine more complex interaction relationships between items, effectively reducing the loss of original information caused by the conversion of multivariate relationships into binary relationships. At the same time, the global interest item transition graph is constructed to combine the global sessions and model the transition relationship between items in adjacent sessions. In the construction process of two graphs, we add an extra interest node for each session, and connect the interest node with each item node in the corresponding session, so that the information between non-adjacent items can be better propagated. After learning by different graph neural networks, we use contrastive learning to enhance the consistency of interest embedding in the two graphs, so that the information in the two graphs complements each other. After fusing the item representations of the two graphs, we design the position and time awareness attention layer to obtain the session embedding by fusing the position information and time information and assigning weights to different items. And we make recommendations based on that.

Our main contributions in this paper are listed as follows :

- We propose a novel model for session-based recommendation named Interest aware Dual-channel Graph Contrastive learning (IDGC), including a dual-channel interest-item embedding learning module to dig the higher-order relationships between items and users' interests in two graphs.
- In the process of constructing graphs, we generate the interest node for each session to represent the users' interest preferences and integrate the users' interest awareness obtained from different views through contrastive learning.

- To refine the session representation, and better mine the limited information in the session, session embedding is enhanced by explicitly injecting the time interval information between items and the position information of items.

In order to verify the effect of the IDGC model, we conducted a series of experiments on three benchmark datasets. The experimental results show that IDGC significantly outperforms the state-of-the-art in the session-based recommendation, proving the effectiveness of the model.

2 Related Work

The main idea of the early SBR(session-based recommendation) model is to explore the relationship between items. For example, the Markov chain-based model [3] uses Markov chain to model the item transition. ItemKNN [1] uses the co-occurrence frequency between items to calculate the similarity of items for recommendation. However, these models often fail to extract the feature information contained in the session sequence.

With the development of deep learning, Recurrent Neural Network (RNN) has played a better role in processing data with sequential information and has been applied to SBR. GRU4REC [5] uses multiple stacked Gated Recurrent Unit (GRU) modules to model interactions between items. After that, a series of improved RNN-based session recommendation was proposed [7,9,13].

In order to model more complex item transition relationships, Graph Neural Network (GNN) has been applied to SBR, and the dependencies between items have been upgraded from point dependencies to collective dependencies. SRGNN [15] introduces graph neural network into SBR for the first time, uses Gated Graph Neural Network (GGNN) to model items. SGNN [11] creates a star graph neural network, which adds a star node to each session when creating a session graph.

However, the above models are not applied to other sessions related to the current session. GCE-GNN [14] refers to the GAT mechanism, uses the current session and all other session information to learn the representation of items. MSGIFSR [4] proposes a multi-granularity consecutive user intent learning model, which captures the interaction between different granularity intent units.

In addition, many different auxiliary information is applied to session-based recommendation. CoHHN [18] combines price information and designs a heterogeneous hypergraph to extract users' preferences. But these works are still difficult to deal with the complex many-to-many relationship between items.

Compared with ordinary graph structures, hypergraphs can better model multivariate relationships due to the property that hyperedges can connect multiple vertices.DHCN [17] constructed a dual-channel hypergraph convolutional network and fused two session representations through self-supervised learning. HIDE [8] constructs hyperedges for the transition between items through distinct perspectives, and disentangles the intents in micro and macro manners. However, the existing work based on hypergraph does not have a good representation of user interest and focuses more on the learning of item embedding.

3 The Proposed Model

In this section, we first introduce the construction of interest item interaction hypergraph and global interest item transition graph. Then we show the propagation and update between interest node and item node, and generate the session embedding. Finally, contrastive learning is used to enhance the model Fig. 1.

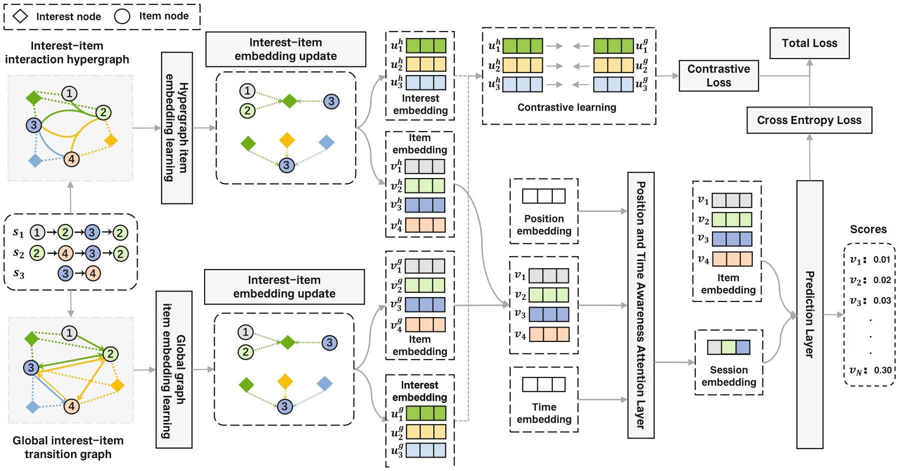


Fig. 1. The overview of the proposed model IDGC. We first build the interest item interaction hypergraph and the global interest item transition graph, and add the interest node for each session in the graph, hypergraph convolutional network and gated graph neural network are used to process the two graph structures respectively. After dual-channel interest-item embedding learning, contrastive learning is performed on the learned interest embedding, and the learned item embedding is aggregated through the position and time awareness attention layer session embedding, and making predictions.

3.1 Notations and Definitions

Given $V = \{v_1, v_2, ..., v_{|V|}\}$ as the item set, where $|V|$ is the number of items. Given $S = \{s_1, s_2, ..., s_{|S|}\}$ as the session set, where $|S|$ is the number of sessions. An anonymous session is represented as a set $s = [v_1, v_2, ..., v_n]$, where v_i represents the i - th interaction item in session s, and n represents the length of session s, the goal of session-based recommendation is to predict the next click v_{n+1}.

3.2 Graph Model Construction

Interest-Item Interaction Hypergraph Construction. Interest-item interaction hypergraph uses undirected hypergraph to model the user's interest nodes and item nodes, and pays more attention to the high-order connections between items.

Define $G_h = (V_h, U_h, E_h, W_h)$, where V_h represents all items in the item set V, U_h represents the interest nodes corresponding to each session. We add an **interest node** for each session to learn the users' long-term interest as a whole. $E_h = (E_h^1, E_h^2)$ represents the set of edges, where E_h^1 represents the set of hyperedges, the i-th hyperedge $e_i^h \in E_h^1$ connects all item nodes including in i-th session; E_h^2 represents the set of undirected edges connecting the interest nodes corresponding to the session and all items in the same session. W_h represents the hyperedge weight matrix. In addition, the hypergraph can be represented as an association matrix $C_h \in R^{|V| \times |S|}$, where $C_h^{v,e} = 1$ if the hyperedge $e \in E_h^1$ contains the vertex $v \in V_h$, otherwise 0.

Global Interest-Item Transition Graph Construction. The global interest-item transition graph uses a directed graph structure to focus on the transition and transmission between item nodes and interest nodes.

Define $G_g = (V_g, U_g, E_g)$, where V_g represents all items in the item set V, U_g represents the interest nodes corresponding to each session. $E_g = (E_g^1, E_g^2)$ represents the set of edges. E_g^1 is the directed edge formed according to the order of interaction between item nodes. If in the whole session set, the user clicks v_i after clicking v_{i-1}, then $e_i = (v_{i-1} \rightarrow v_i) \in E_g^1$. E_g^2 is the undirected edge connecting an interest node with an item node. We use the adjacency matrix A_s to represent G_g. The construction method of the adjacency matrix A_s is to concatenate the outgoing matrix and the incoming matrix of the graph.

3.3 Dual-Channel Interest-Item Embedding Learning

In this section, we introduce the dual-channel interest-item embedding learning module, and design two channels to process the constructed interest item interaction hypergraph G_h and global interest item transition graph G_g respectively.

Hypergraph Item Embedding Learning. We use hypergraph convolutional network [2,17] to learn hypergraph item embedding at the item level. The specific implementation method is as follows :

$$\begin{aligned} H_h^{l+1} &= D^{-1} C B^{-1} C^T H_h^l \\ H_h &= \frac{1}{L+1} \sum_{l=0}^{L} H_h^l \end{aligned} \tag{1}$$

where H_h^l represents item embedding in the l-th layer hypergraph convolutional network, and the range of l is 0 to L. C is the association matrix of interest item interaction hypergraph G_h mentioned in Sect. 3.2. $D = \sum_{e \in E_h^1} w(e) C_h^{v,e}$ and $B = \sum_{v \in V_h} C_h^{v,e}$ stand for the degree matrix of the hyperedges in E_h^1 and the degree matrix for all the vertices in V_h respectively.

After stacking multiple hypergraph convolutional networks to obtain high-level information, we output the embedding representation of each node, and finally get the item embedding H_h under the interest item interaction hypergraph.

Global Graph Item Embedding Learning. In order to learn global graph item embedding, we input the constructed global interest item transition graph G_g into the Gated Graph Neural Network (GGNN) [15], as follows:

$$\begin{aligned}
a_i^l &= A_s[h_1^{g,l-1}, h_2^{g,l-1}, h_3^{g,l-1}, ..., h_{|V|}^{g,l-1}]^T U_1 + b_1 \\
z_i^l &= sigmoid(U_2 a_i^l + W_1 h_i^{g,l-1}) \\
r_i^l &= sigmoid(U_3 a_i^l + W_2 h_i^{g,l-1}) \\
\tilde{h}_i^{g,l} &= tanh(U_4 a_i^l + W_3(r_i^l \odot h_i^{g,l-1})) \\
h_i^{g,l} &= (1 - z_i^l) \odot h_i^{g,l-1} + z_i^l \odot \tilde{h}_i^{g,l}
\end{aligned} \tag{2}$$

where $h_i^{g,l}$ represents i-th item embedding in the l-th layer network, and the range of l is 0 to L. When $l = 0$,A_s is the adjacency matrix, $U_1, U_2, U_3, U_4 \in R^{d\times 2d}$ and $W_1, W_2, W_3 \in R^{d\times d}$ control the weights, $b_1 \in R^d$ is a trainable parameter, and $\odot$ is the elementwise multiplication operator.

Through the global graph item embedding learning module, we finally get the item embedding H_g under the global interest item transition graph.

Interest-Item Embedding Update. After obtaining item embedding, we obtain item embedding and interest embedding updated under a certain view through multi-layer propagation and update.

Take interest item interaction hypergraph as an example, for each session $s = [v_1, v_2, ..., v_n]$, the interest node corresponding to s is u^h in interest item interaction hypergraph. we initialize the interest node embedding as follows:

$$u^{h'} = \frac{1}{n}\sum_{i=1}^{n} h_i^h \tag{3}$$

$$\alpha_i = q_1^T (W_4 h_i^h + W_5 u^{h'} + b_2) \tag{4}$$

$$u_f^h = \sum_{i=1}^{n} \alpha_i h_i^h + h_n^h \tag{5}$$

where h_i^h is item embedding under interest item interaction hypergraph.u_f^h is the initialized interest node embedding. $W_4, W_5 \in R^{d\times d}$ and $q_1, b_2 \in R^d$ are all learnable parameters.

After initializing the interest nodes, we update the interest embedding according to the obtained item embedding.

$$u^h = \sum_{i=1}^{n} softmax(LeakyReLU(W_6 h_i^h + W_7 u_f^h)) h_i^h \tag{6}$$

where $W_6, W_7 \in R^{1\times d}$are all learnable parameters.

Similarly, we get the updated embedding u^g of the interest node in the global interest item transition graph.

After updating the interest embedding, we use the updated interest node to update the item node.

$$h_f^h = \sum_{j=1}^{k} softmax(LeakyReLU(W_8 h^h + W_9 u_j^h))u_j^h \tag{7}$$

where k represents item node is connected to k interest nodes. $W_8, W_9 \in R^{1\times d}$are all learnable parameters.

Similarly, we get the updated embedding h_f^g of the item node in the global interest item transition graph. We add the item embedding h_f^h and h_f^g obtained from the two views to get the final item embedding $H = [h_1, h_2, h_3, ..., h_{|V|}]$.

3.4 Generating Session Embedding

After obtaining the item embedding that fuses the two graph information, in order to get the session embedding, we designed the position and time awareness attention layer, and added the position information and time information in the process of generating the session embedding.

In session-based recommendation, the same item appears in different positions in the session contains different information. Therefore, in each session s, we use a learnable position matrix $P = [p_1, p_2, p_3, ..., p_n]$ to record the position information of items in each session.

It can be intuitively felt that without changing the order of interaction between items, two sessions with different time intervals represent completely different user intentions. We obtain the time matrix $T = [t_1, t_2, t_3, ..., t_n]$ that records the time information of different items by calculating the difference of timestamps between adjacent items.

For each session, we add P and T into item embedding H, get the final item embedding $X = [x_1, x_2, x_3, ..., x_n]$. In general, we can think that the later the item is viewed, the more it can represent the user's current interest preference, so we use the last item embedding x_n in session s as local-level session embedding, and then use soft attention mechanism to compute the weight to obtain global-level session embedding. Combine the global-level embedding of the session with the local-level embedding to get the session embedding s_f.

$$s^* = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8}$$

$$\theta_i = q_2^T(W_{10}x_i + W_{11}s^* + b_3) \tag{9}$$

$$s = s_l + s_g = x_n + \sum_{i=1}^{n} \theta_i x_i \tag{10}$$

where $W_{10}, W_{11} \in R^{d\times d}$ and $q_2, b_3 \in R^d$ are all learnable parameters

3.5 Contrastive Learning

Through propagation and update, we get the representation of interest nodes in two different views. Intuitively, the interest preferences of the same session in two different views should be as similar as possible. Since the total number of sessions defined in session set S is $|S|$, the total number of all interested nodes in the two views is $2|S|$. We define two interest embeddings (u_j^h, u_j^g) that represent the interest node of the same session in two views as a pair of positive samples, and the remaining $2|S| - 1$ pairs are negative samples. We use InfoNCE loss [10] as a contrast loss to optimize the model so that the encoding of interest nodes in the same session in both views is as similar as possible.

$$\mathcal{L}_{\int} = \frac{1}{|S|} \sum_{j=1}^{|S|} -log(\frac{exp(sim(u_j^h, u_j^g)/\tau)}{\sum_{k=1}^{2\,S|} exp(sim(u_p, u_q)/\tau)}) \tag{11}$$

where (u_p, u_q)denotes all pairs of sample, $sim(\cdot)$ denotes the cosine similarity, τ is a hyper-parameter.

3.6 Making Recommendation and Model Training

After obtaining item feature representation and session feature representation, we then obtain the final recommendation probability y_i of the i-th item as below.

$$\hat{y}_i = softmax(s * h_i) \tag{12}$$

In order to optimize the model, we employ the cross entropy function as the loss function, then we add the contrastive loss $\mathcal{L}_{\int}$ mentioned in Sect. 3.5 to the cross entropy loss to get the total loss $\mathcal{L}$.

$$\mathcal{L}_{\rfloor} = -\sum_{i=1}^{|V|} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{13}$$

$$\mathcal{L} = \mathcal{L}_{\rfloor} + \lambda \mathcal{L}_{\int} \tag{14}$$

where y_i is the ground truth label of i-th item and $\hat{y}_i$ is the prediction results of i-th item, λ is a hyper-parameter.

4 Experiments

4.1 Experimental Settings

In order to verify the effectiveness of the IDGC model, we selected three datasets commonly used in session-based recommendation algorithms to verify our model, which are Diginetica[1], Yoochoose 1/4[2] and Yoochoose 1/64[3] Because

[1] http://cikm2016.cs.iupui.edu/cikm-cup.

[2] http://2015.recsyschallenge.com/challenge.html.

[3] http://2015.recsyschallenge.com/challenge.html.

Table 1. Statistics of the experimental dataset

Dataset	**Clicks**	**Train sessions**	**Test sessions**	**Items**	**Average length**
Diginetica	982961	719470	60858	43097	5.12
Yoochoose 1/4	8326407	5917745	55898	29618	7.42
Yoochoose 1/64	557248	369859	55898	16766	6.16

of Yoochoose dataset's large scale, we refer to the practice of SRGNN [15] and other models, and use the most recent 1/4 data and the most recent 1/64 data of Yoochoose training set to form two datasets: Yoochoose 1/4 and Yoochoose 1/64. For fairness, we use the same data processing method as SRGNN [15] to process the datasets. The statistics of the processed dataset are displayed in Table 1.

Evaluation Metrics. We use Precision@20 (P@20) and Mean Reciprocal Ranking@20 (MRR@20) to evaluate the effect of the model, P@20 represents the proportion of predicted correct results to all returned results, MRR@20 represents the mean value of the reciprocal of the correct recommended item priority ranking.

Hyper-parameters Settings. We set 0.1 of the training set as the validation set, the batch size to 1024, and the hidden size to 256. For the optimizer, we choose to use the Adam optimizer and set an initial learning rate $1e^{-3}$ and a decay factor of 0.1 for three epochs. In addition, the hyperparameter τ in contrastive learning is set to 0.2, and the fusion ratio λ is 0.1.

Table 2. The performance of IDGC vs. other baseline models on 3 benchmark datasets (all values are in percentages).

Dataset	**Diginetica**		**Yoochoose 1/4**		**Yoochoose 1/64**	
	P@20	**MRR@20**	**P@20**	**MRR@20**	**P@20**	**MRR@20**
Item-KNN [1]	35.75	11.57	52.31	21.70	51.60	21.81
FPMC [12]	26.53	6.95	–	–	45.62	15.01
GRU4Rec [5]	29.45	8.33	59.53	22.60	60.64	22.89
NARM [7]	49.70	16.17	69.73	29.23	68.32	28.63
STAMP [9]	45.64	14.32	70.44	30.00	68.74	29.67
SR-GNN [15]	50.73	17.59	71.36	31.89	70.57	30.94
COTREC [16]	54.04*	18.72*	71.48*	31.19*	70.72*	30.36*
Disen-GNN [6]	53.79	18.99	–	–	71.46	31.36
GCE-GNN [14]	54.22	19.04	71.61*	31.49*	71.91*	30.83*
IDGC	**54.98**	**19.05**	**72.74**	**32.67**	**71.96**	**32.02**

4.2 Experimental Results

We show the experimental results of overall performance in Table 2, where we mark the best results in a column by thickening. The scores of the tag * represent the result of our reimplementation according to the open source code because of the different datasets used. According to the experimental results, we get some conclusions :

Compared with the traditional model (item-KNN, FPMC), the effect of the RNN-based model (GRU4REC, NARM, STAMP) using sequence information is greatly improved. This result shows that considering the location information between items, the method of modeling session information as a sequence is effective. The overall effect of the GNN-based model is better than the traditional model and the RNN-based model, because modeling the session data into a graph can better represent the more complex links between items.

Our proposed IDGC model outperforms all baselines under all evaluation metrics on all datasets, which proves the effectiveness of our model. The improvement of IDGC on the Yoochoose 1/4 dataset is significantly better than that of other datasets, which we believe is because the Yoochoose 1/4 dataset has more training data, so the model can learn more abundant high-order relationships between items in the process of building the graph. This demonstrates the strong learning ability of our model.

4.3 Ablation Study

To investigate the contribution of each module to the model, we conduct a series of ablation experiments.

Table 3. Impact of each module in ablation study(all values are in percentages).

	Nhyperg	Nglobalg	Nssl	Npos	Ntime	Npos&time	IDGC
Diginetica(P@20)	54.60	54.69	54.85	54.87	54.85	54.75	54.98
Diginetica(MRR@20)	18.79	18.82	18.87	18.93	18.90	18.87	19.05
Yoochoose 1/4(P@20)	72.38	72.32	72.54	72.49	72.65	72.28	72.74
Yoochoose 1/4(MRR@20)	32.39	32.47	32.45	32.50	32.33	31.98	32.67
Yoochoose 1/64(P@20)	71.71	71.85	71.88	71.83	71.88	71.76	71.96
Yoochoose 1/64(MRR@20)	31.84	31.93	31.96	31.89	31.91	31.74	32.02

We focus on different aspects of the relationship between items by constructing two different graph structures. So we designed two versions: Nhyperg and Nglobalg, in the Nhyperg version, the interest item interaction hypergraph is deleted, and the global interest-item transition graph module is deleted in the Nglobalg version. The experimental results are shown in the Table 3. It can be seen that the experimental results of the Nglobalg version have declined but not as much as the Nhyperg version, which proves that the interest item interaction hypergraph has a stronger ability to characterize complex item relationships than the global interest-item transition graph.

In order to verify the effectiveness of contrastive learning, we designed the Nssl version and deleted the contrastive learning module. The experimental results are shown in Table 3. We can find that the experimental effect has been reduced. It can be seen that the decline in Nssl in the Yoochoose 1/64 dataset is smaller than that in other datasets. We think this is because there are fewer training sessions in Yoochoose 1/64, so the contrastive learning module can learn less user preference information, which leads to a smaller drop in the effect of removing the contrastive learning module.

In addition, we designed three versions: Npos, Ntime, and Npos&time, which represent the versions that does not consider position information, do not consider time information, and only uses soft attention mechanism to aggregate item. The results are shown in Table 3. It can be found that the effect of Npos&time is worse than that of Npos and Ntime, which proves the effectiveness of increasing position information and time information.

5 Conclusion

In this paper, we propose a novel model Interest aware Dual-channel Graph Contrastive learning (IDGC) for session-based recommendation. We complete two graphs with different structures to focus on different aspects of information between items and generate interest nodes to improve the long-distance information propagation between non-adjacent items. The interest embedding of the two graphs is enhanced by contrastive learning strategy to reduce the influence of data sparsity and noise. Meanwhile, we incorporate position information and time information when aggregating the session representation to better mine the limited information. We conduct extensive experiments on three benchmark datasets, and the experimental results show the effectiveness and necessity of our model.

References

1. Davidson, J., et al.: The youtube video recommendation system. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 293–296 (2010)
2. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3558–3565 (2019)
3. Gu, W., Dong, S., Zeng, Z.: Increasing recommended effectiveness with Markov chains and purchase intervals. Neural Comput. Appl. **25**, 1153–1162 (2014)
4. Guo, J., et al.: Learning multi-granularity consecutive user intent unit for session-based recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 343–352 (2022)
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)

6. Li, A., Cheng, Z., Liu, F., Gao, Z., Guan, W., Peng, Y.: Disentangled graph neural networks for session-based recommendation. IEEE Transactions on Knowledge and Data Engineering (2022)
7. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428 (2017)
8. Li, Y., Gao, C., Luo, H., Jin, D., Li, Y.: Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1997–2002 (2022)
9. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1831–1839 (2018)
10. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
11. Pan, Z., Cai, F., Chen, W., Chen, H., De Rijke, M.: Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management,d pp. 1195–1204 (2020)
12. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World wide web, pp. 811–820 (2010)
13. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning For Recommender Systems, pp. 17–22 (2016)
14. Wang, Z., Wei, W., Cong, G., Li, X.L., Mao, X.L., Qiu, M.: Global context enhanced graph neural networks for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 169–178 (2020)
15. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 346–353 (2019)
16. Xia, X., Yin, H., Yu, J., Shao, Y., Cui, L.: Self-supervised graph co-training for session-based recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2180–2190 (2021)
17. Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4503–4511 (2021)
18. Zhang, X., et al.: Price does matter! modeling price and interest preferences in session-based recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1684–1693 (2022)

Chinese Event Causality Identification Based on Retrieval Enhancement

Yumiao Gao, Yizhi Ren, Jiawei Rao, Zuohua Chen, Qisen Xi, Haoda Wang, Dong Wang, and Lifeng Yuan(✉)

School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
{gaoyumiao,renyz,rjw,chenzuohua,xiqs,wanghaoda, wangdong,yuanlifeng}@hdu.edu.cn

Abstract. Event causality identification (ECI) is a critical and challenging information extraction task, which aims to identify whether there is a causal relationship between the two events. To address the current problem of insufficient annotated data in Chinese event causality identification, and the rich semantics of the event itself is not exploited. We proposed a prompt-learning based retrieval enhancement Chinese events causality identification framework, named RE-CECI, which improves the few-shot learning capability of the model. Moreover, it enables the pre-trained language model to better learn event information by adding retrieved examples to the input text, after which the model can learn more event information. In addition, we construct a retrieval store of retrieved examples to serve as clues for reasoning about causality. Our experimental results on the only available Chinese causality dataset, show that our proposed method significantly improves the performance of Chinese event causality identification.

Keywords: Event causality identification · Pre-trained model · Prompt-learning

1 Introduction

Event causality identification, as a prominent research direction in the field of natural language processing, aims to discern the presence of a causal relationship between two events. This task has diverse applications in various downstream NLP tasks, such as information extraction [1] and multi-turn question answering [2]. It also plays an important role in the field of artificial intelligence [3]. However, research on Chinese event causality identification is not yet sufficient. Currently, the state-of-the-art approach for Chinese event causality identification typically involves employing improved neural network models. The most commonly used approach is the Attention-BiGRU network-based [4] method, which achieves the highest accuracy and F1 scores compared to traditional pattern matching and other neural network approaches. Despite the progress made in using neural network models for event causality identification, there are still challenges, particularly in the Chinese language domain. The existing event causality

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 159–170, 2023.
https://doi.org/10.1007/978-3-031-44693-1_13

datasets in Chinese are often limited in terms of labeled data, resulting in poor generalization performance of models trained with traditional methods. Such models tend to perform well on training data but poorly on unseen data, leading to issues with overfitting.

It is evident that the insufficient labeled data poses challenges to the performance of event causality identification models, and there is a need to enhance the few-shot learning capability of the model. Note that the currently popular approach of prompt learning [5], which formulates the original task as a language model problem, by incorporating natural language prompts and a few demonstration examples into the input text. This approach has shown great results in few-shot or zero-shot scenarios. There is also current work that introduces prompt learning into ECI tasks. For example, Shen et al. [6] proposed two derivative prompt tasks from event causality identification to perform causality identification. However, there are no studies exploring the role of demonstration examples for ECI. Demonstration examples, as a critical component of prompt learning, play a crucial role in improving the performance of language models by training them to recognize diverse patterns [7].

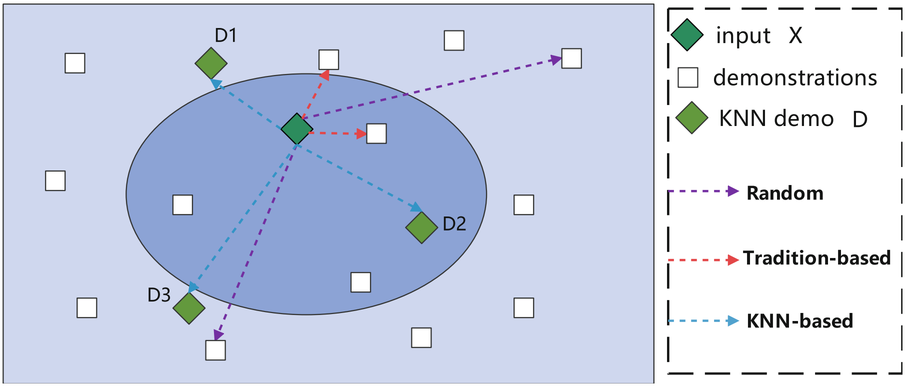


Fig. 1. Comparison of current demonstration examples sampling strategies. Compared with random search and traditional search, KNN lookup can obtain a better distribution of demonstration examples.

Therefore, we propose a retrieval-enhanced method for Chinese event causality identification. Our approach involves constructing a retrieval store for demonstration examples and utilizing K-nearest neighbor (KNN) lookup for retrieval. As illustrated in Fig. 1, Compared to random retrieval or traditional retrieval, the examples retrieved using KNN can undoubtedly facilitate the learning of event representations. Afterwards, using the retrieved examples serve as clues for causality identification and enhances the model's effectiveness, especially when faced with low resources. Additionally, we introduce a **R**etrieval-**E**nhanced **C**hinese **E**vent **C**ausality **I**dentification framework named RE-CECI, which concatenates the retrieved examples with the input text and prompt. The input of the model is constructed as "$[SEP] < Example > [SEP] < Sentence > [SEP] < Prompt > [SEP]$". The causality prediction is performed using the pre-trained model with the retrieved examples and prompts as contexts. This enables effective handling of the ECI task with only a small amount of labeled data.

The contributions of our method are as follows:

(1) We propose a novel retrieval-enhanced framework for Chinese event causality identification, called RE-CECI. This framework employs retrieved demonstration examples to enhance the performance of event causality identification.
(2) Through detailed ablation experiments, we have proved the effectiveness of the demonstration examples in enhancing the performance of ECI. By evaluating RE-CECI on a Chinese causality dataset, we achieved the state-of-the-art performance to date.
(3) Through few-shot learning experiments, we have proved the effectiveness of prompt-learning in improving the few-shot capability of the ECI model, thus mitigating the issue of insufficient labeled data.

2 Related Work

Event causality identification is a prominent research area in natural language processing (NLP) that has gained significant attention in recent years. Its aim is to determine whether a causal relationship exists between two events. Initially, rule-based methods were used to rely on pattern matching, such as constructing lexical and syntactic patterns [8] or summarizing causal patterns based on algorithmic approaches [9], which were commonly employed for causality identification. Subsequently, many approaches have combined pattern matching with machine learning techniques [10] to enhance the model's effectiveness, by incorporating lexical and grammatical features as inputs for event causality identification [11].

With the advancements in deep learning and the remarkable learning capabilities of neural networks [12], the utilization of deep learning techniques for causal relationship identification has gained prominence in recent years. Ponti et al. [13] pioneered the use of neural network models for event causality identification, in addition to basic lexical features, also incorporating positional features. De Silval et al. [14] employed WordNet knowledge base and CNN to complete the identification of causal relationships. While Eugenio et al. [15] utilized LSTM (Long Short-Term Memory) models for training and achieved state-of-the-art results on the Penn Discourse TreeBank (PTDB) dataset at that time.

With the emergence of BERT (Bidirectional Encoder Representations from Transformers) [16], pre-trained language models have been increasingly employed in various natural language processing (NLP) areas, including event causality identification (ECI). Wu et al. [17] introduced entity information based on pre-trained BERT encoding and connected BERT-based features of each word with entity pair representations for event causality identification. Kyriakakis et al. [18]

explored the performance of transfer learning using BERT for causal knowledge discovery and conducted comparative experiments with ELMO, indicating that the improvements are particularly significant on small causal datasets. Liu et al. [19] proposed a knowledge enhancement approach combined with a mention mask generalization mechanism for causality identification.

3 Methodology

In this section, we introduce our innovative framework, Retrieval Enhanced Chinese Event Causality Identification (RE-CECI), as illustrated in Fig. 2. Specifically, the proposed model leverages a retrieval store to retrieve examples that serve as clues for event causality identification, each example is stored as $< example, result >$ pairs in the retrieval store, and then encoded into the input to enable the pre-trained model to better learn relevant task-specific information. The framework consists of four key components: retrieval store construction, example retrieval, example embedding, and event causality identification, which we elaborate on below. Before to delving into the framework, we provide a necessary introduction to the prompt-tuning-based approach to enhance understanding of our proposed method. Lastly, we detail the training and prediction process of the model.

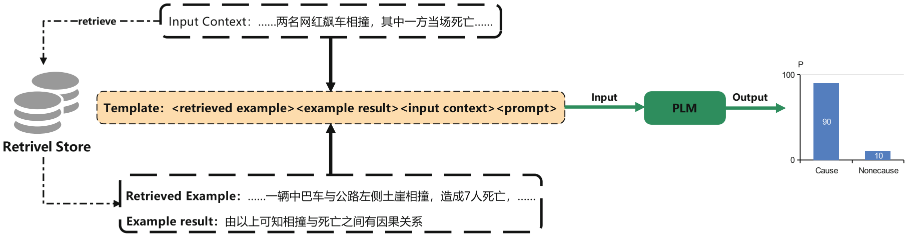

retrieve
Input Context: ......两名网红飙车相撞，其中一方当场死亡......
Retrivel Store
Template: <retrieved example><example result><input context><prompt>
Input
PLM
Output
P
100
0
90
10
Cause
Nonecause
Retrieved Example: ......一辆中巴车与公路左侧土崖相撞，造成7人死亡，......
Example result: 由以上可知相撞与死亡之间有因果关系

Fig. 2. Retrieval-based enhanced event causality identification model framework.

3.1 Prompt-Tuning of PLMS

Most pre-training models(PLM) in the past have relied on fine-tuning to perform downstream tasks. While this approach can yield satisfactory results, there remains a disparity between the downstream task and the pre-training task. To address this gap, the prompt-tuning approach was proposed, It aims to avoid introducing additional parameters by adding prompt, so that the language model can achieve the desired results in Few-shot or Zero-shot scenarios. Prompt-tuning execution includes two steps: Template Construction and Label Word Verbalizer. To construct the prompt, we begin by creating a template that contains *[MASK]* token which is then transformed to a format $x_t = T(x)$, where x is an instance of the input, T is Template function. Additionally, we create a mapping of labeled words to result words: $v(\cdot)\colon v(y) \rightarrow V$, where y is the output word of the mask label, and V is Label words for answers, v is Verbalizer function. Once these two functions have been established, we input to the *PLM* and obtain a hidden vector $h_{[MASK]}$ at *[MASK]* position. M is PLM, The final predicted label probability distribution is then calculated as follows:

$$P(y\big|x) = P_M([MASK] = v(y)\big|T(x)) \tag{1}$$

3.2 Retrieval Store Construction

We construct the retrieval store with the text set C in the training set, and each data in it is stored as a dictionary. Where the key is the embedding vector h_i for each sample i in the training set, and the value denotes whether there is a causal relationship r_i for this sample i, $r_i \in \{Cause, Nonecause\}$. The overall representation is as follow: $(K, V) = \big\{(h_i, r_i)\big|i \in C\big\}$, we use the model symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli[1] to obtain the embedding vector for each sample. the vector obtained by this model contains rich semantic information, especially in Chinese.

3.3 Example Retrieval

For example, we first retrieve them from the retrieval store, and the examples should be semantically similar to the input text so that they can give clues to the model for causal reasoning. An example is a (K, V) pair, i.e., an $< example, result >$ pair, and we use KNN to find the examples in the retrieval library and calculate the similarity of the samples by the following formula:

$$sim(h_i, h_x) = \frac{h_i^T \cdot h_x}{||h_x|| \cdot ||h_i||} \tag{2}$$

where h_x is the embedding vector of the input text, h_i^T is the transposed vector of h_i, $|| \cdot ||$ denotes absolute value.

3.4 Example Embedding

For the retrieved examples we take the approach of directly concatenating them with the input text and prompt, splicing $< example >$ and $< result >$ with the input text and prompt, separated by a separator, as the input of the pre-trained model. The specific constructed templates are as follows:

$$\begin{gathered} f_T(x, i, r) =< demon > i < /demon >< demonresult > r < /demonresult > \\ < s > x < /s > In\ this\ sentence, \text{“}e_s\text{”}[MASK]\text{“}e_t\text{”} < /s > \end{gathered}$$

[1] http://huggingface.co/symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli.

where x denotes an input instance, $x = (S, e_s, e_t)$, S denotes a sentence, and (e_s, e_t) denotes an event pair in S. i denotes an example of x, and the extraction and representation of i, what is described in the previous subsection. r denotes the result of i, specifically the result of whether i has a causal relationship. e_s represents the source event, and e_t represents the target event. *[MASK]* denotes masking tagging for answer output and constructs the verbalizer as $v(\cdot)$: $v(y) \rightarrow V$, $V \in \{Cause, Nonecause\}$, mapping causal tags to tagged words with the same name. This allows us to capture the relationship between the input text, prompt, and examples in a unified framework, and to leverage the semantic information from the examples to enhance the performance of event causality identification.A concrete template is shown in Fig. 3 below. After transforming the example, input text and prompt into template form, we can see that there is a special token *[MASK]*, which is used as our instance in the embedding vector of the last layer output in representation:

$$h_x = M(x_{template})_{[MASK]} \tag{3}$$

where M is the PLM, x is the input text, and $x_{template}$ is the template we constructed above, input pre-trained model that predicts the probability of the final $P(y|x)$.

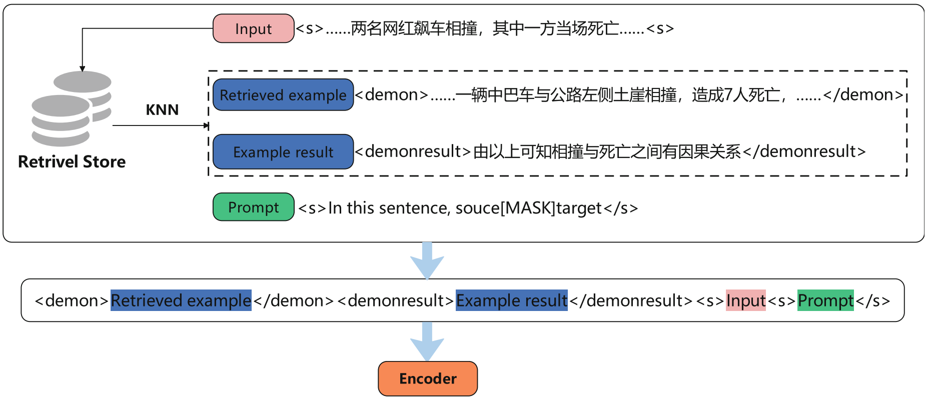

Input
<s>......两名网红飙车相撞，其中一方当场死亡......<s>
Retrieved example
<demon>......一辆中巴车与公路左侧土崖相撞，造成7人死亡，......</demon>
KNN
Retrivel Store
Example result
<demonresult>由以上可知相撞与死亡之间有因果关系</demonresult>
Prompt
<s>In this sentence, souce[MASK]target</s>
<demon>Retrieved example</demon><demonresult>Example result</demonresult><s>Input<s>Prompt</s>
Encoder

Fig. 3. Construction of prompt tuning with demonstrations.

3.5 Event Causality Identification

The current definition of event causality mostly follows the approach of Ikuta et al. [20]. Viewing event causality identification as a text classification problem, given an ECI task instance $x = (S, e_s, e_t)$, where S is denoted as a sentence and (e_s, e_t) is denoted as an event pair in S. For instance x, the purpose of the ECI model is to predict whether there is a causal relationship between e_s

and e_t, where for the output of the ECI model, we follow the traditional binary classification method. That is, the output corresponds to the probability of the causal label $y \in Y$, which is the set of causal labels of whether there is a causal relationship between the event pairs, $V: \{Cause, Nonecause\}$, which denote the existence of a causal relationship between e_s and e_t, and the absence of a causal relationship between e_s and e_t, respectively.

3.6 Training and Prediction

We perform supervised training on the ECI task and calculate the loss of the task using a cross-entropy function with the loss function shown below:

$$\pounds_{[MASK]} = -\frac{1}{|x|}\sum_{x\epsilon X} y \log p(y|x) \tag{4}$$

where x is the input text, X is the set of training sets, and y is the corresponding causal label. In the prediction phase, the target position of the ECI task is the *[MASK]* token, and we predict the probability distribution of the tagged words by Eq. (1).

4 Experiments

4.1 Dataset and Experiment Setup

We evaluated our RE-CECI framework on Chinese Corpus of Emergent Events (CEC), the only Chinese event causality corpus currently available, which consists of five categories of corpus containing a total of five event relations, uses the XML language as the annotation format. For a fair comparison, we followed the same setup as in previous work [21]. Documents were randomly divided in a ratio of 7:1:2 for the train/dev/test set. Event causality identification was used as a text classification task using the same criteria used for classification tasks, i.e., precision (P), recall (R), and F1 score (F1). The pre-training model we used is xml-roberta-base, which consists of 12 layers, 768 hidden units, and 12 attention heads. Adam is chosen as the optimizer, the batch size is set to 16, and the learning rate is 1e-5. The experimental code is based on PyTorch, and the experiments are run on an NVIDIA A800 GPU.

4.2 Main Result

We compared our approach with the following state-of-the-art methods for event causality identification in Chinese: **LSP-based** event causality identification [22] uses a statistical machine learning approach; **CSNN** [23] constructs a cascaded structural model that combines CNN and LSTM with a self-attentive mechanism for causality identification; **MCBi-GRUs** [21] combines the advantages of both convolutional neural networks and recurrent neural networks to build a multi-column convolutional bidirectional gated recurrent unit neural network model;

BiLSTM-TWAM+CRF model [24] captures more text features by introducing external vocabulary and using the constructed two-way attention module TWAM; **Att-BiGRU** [25] proposes a joint word vector and context-generated dynamic word feature model that uses a combination of BiGRU networks and attention mechanisms to extract event causality. Furthermore, for a fair comparison, we build two baseline models based on RoBERTa: (1) **RoBERTa-base**: we use a linear classifier after RoBERTa for the event causality identification task, and the input of the classifier is the hidden features of the target event. (2) **Prompt-base**: the prompt-based approach i.e., the prompt-tuning approach using the pre-trained model we described in Sect. 3.1. Meanwhile, we evaluated the performance of **ChatGPT** on the Chinese event causality identification task.

Table 1. The overall performance on Chinese event causality identification. Bold denotes the best results.

Model	**P**	**R**	**F1**
Based on LSP	76.5	82.3	79.3
CSNN	70.6	54.9	61.7
MCBi-GRUs	80.2	85.9	82.9
BiLSTM-TWAM+CRF	76.4	71.9	74.0
Att-BiGRU	92.2	85.2	88.6
RoBERTa-base(ours)	80.3	74.6	78.3
Prompt-base(ours)	83.6	78.3	80.9
ChatGPT-CECI	76.1	68.3	72.0
RE-CECI(ours)	**92.6**	**91.5**	**92.3**

The above Table 1 shows the results of Chinese event causality identification on CEC, which shows that:

(1) Our model has a significant performance improvement, firstly the proposed method in this paper improves the F1 value by at least 13% compared to the statistical machine learning based method. Secondly, the F1 value is improved by at least 10% compared to the neural network-based method. And compared with the current state-of-the-art method Att-BiGRU, our method improves the F1 value by 4%. The results prove the effectiveness of the proposed method in this paper for event causality identification.

(2) The performance of the RoBERTa-based method outperforms most of the methods that do not use RoBERTa, illustrating the superiority of using the RoBERTa method, but RoBERTa-base is inferior to Att-BiGRU, also illustrating that simply fine-tuning the PLM cannot fully capture the knowledge required for the ECI task. The reason why the Prompt-base approach is superior to RoBERTa-base may be that the Prompt-base approach can directly translate the potential causal knowledge of PLM into causal recognition capability.

(3) The poor performance of ChatGPT in Chinese event causality identification may be due to the lack of a clear correct answer during the learning training process, e.g., some cautious training strategies lead to the model's inability to produce correct or the complexity of Chinese language, may lead to lower performance.

4.3 Ablation Study

We also conducted three ablation studies, and the results are shown in Table 2 for the specific model variants as follows: 1) *None retrieval:* by omitting the retrieval store module to downgrade our proposed framework, the F1 value of our model immediately decreases by about 15%, which demonstrates the main reason for the improved performance based on the retrieval enhancement approach. 2) *Retrieval ([random examples]):* we retrieved examples using the random retrieval approach, and the retrieved examples are stitched with the input text by our proposed method for event causality identification, and we find that its performance decreases but increases compared to the no-retrieval method. 3) *Retrieval ([error examples]):* we retrieved examples using the traditional retrieval approach, and the retrieved error examples are stitched with the input text by our proposed method for event causality identification, As anticipated, this caused a slight decrease in model performance. However, compared to not using a retrieval method at all, we still observed an improvement in performance. Thus, we can conclude that the retrieval-based method is beneficial for ECI tasks, even in the case of incorrect examples. The longest input length of the model we use is 512, and when the whole sample is used, it is input separately .

Table 2. Result of the ablation study, where k is the number of samples, Full means full sample, and F1 scores are reported for each evaluation metric.

Method	k=1	k=5	k=16	Full
RE-CECI (our)	25.3	47.8	68.9	93.6
- None Retrieval	20.2	43.4	57.3	78.5
- Retrieval ([random example])	22.4	44.7	65.2	88.6
- Retrieval ([error example])	21.4	44.3	60.2	85.3

4.4 Analysis of Retrieval Similarity Experiment Results

In this section, we studied the effect of different levels of similarity on our model. The test retrieved examples were divided into five intervals according to the cosine similarity, namely $(< 0.5, 0.5 - 0.6, 0.6 - 0.7, 0.7 - 0.8, > 0.8)$, and then we counted the event-cause recognition F1 values in each similarity interval. The results are shown in Fig. 4, we can observe that the F1 values corresponding to the five similarity intervals are 0.8534, 0.8823, 0.9165, 0.9187, 0.9263 respectively. According to the experimental results, it can be observed that when the example similarity increases, the performance of the model is also increasing.

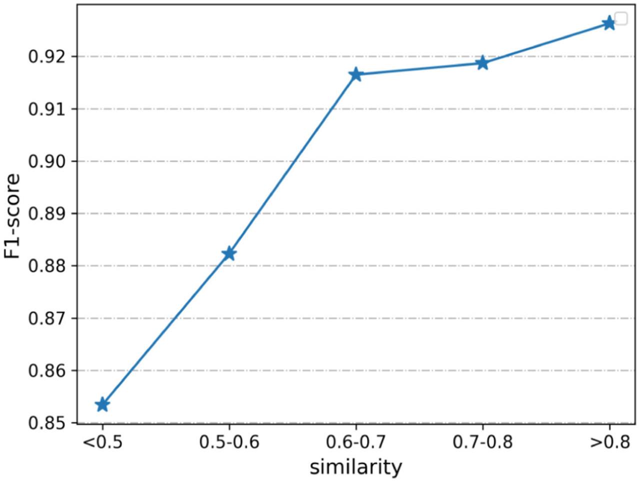


Fig. 4. Experimental results of retrieval similarity Research.

4.5 Few-Shot Study

This section conducts few-shot learning capability experiments, aiming to verify whether the few-shot learning capability of the prompt learning paradigm plays a role in the event causality identification task. The experiments use fine-tuning and prompt-learning paradigm comparisons to compare their performance in the few-shot case. The training set for the few-shot learning ability experiment consists of several strips of data from the original training set. In this experiment, One-Shot, 5-Shots and 16-Shots datasets were selected for few-shot experiments. The training data are selected in a balanced manner, with an approximately equal number of causal and non-causal relations. The experimental results are presented in Table 3.

Table 3. Comparing Few-shot Learning Results.

Method	One-Shot	5-Shots	16-Shots
Fine-tuning	14.3	36.7	52.8
RE-CECI(hard-template)	23.7	44.5	68.4
RE-CECI(soft-template)	25.3	47.8	68.9

It can be found that the highest F1 value with prompt-learning paradigm is increased to 68.9% at 16-Shots only, which is only 23.4% different from the result of Full-Shots experiment we did before, and the F1 value can still reach 47.8% in the case of 5-Shots. This shows that the small-sample learning ability of prompt-learning is still effective in Chinese event causality identification. This can well alleviate the problem of insufficient labeled data in Chinese event causality identification.

5 Conclusions

This paper proposes a retrieval-based enhanced event causality identification framework, RE-CECI. Previous approaches to introduce prompt learning into ECI tasks have not taken into account the importance of demonstration, which does not allow the pre-trained model to learn more information about the task. RE-CECI enhances the context learning ability of the pre-trained model by retrieving the demonstrations to enhance the contextual learning ability of the pre-trained model. Experiments on public benchmark show that our work outperforms previous competitive methods. Our future work will address how to better construct retrieval store and how to retrieve them more efficiently and apply them to general causality identification tasks, such as document-level Chinese causality identification.

Acknowledgement. Zhejiang Province's "Sharp Blade" and"Leading Goose" Resea rch and Development Projects(No.2023C03203, 2023C03180, 2022CO3174),Zhejiang Province-funded Basic Research Fund for Universities Affiliated with Zhejiang Province (NO.GK229909299001-023).

References

1. Wang, J., Lin, C., Li, M.: Boosting approximate dictionary-based entity extraction with synonyms. Inform. Sci. **530**, 1–21 (2020)
2. Romeo, S., Da San, M.G., Belinkov, Y.: Language processing and learning models for community question answering in Arabic. Inform. Process. Manage. **56**(2), 274–290 (2019)
3. Savage, N.: Why artificial intelligence needs to understand consequences. Nature (2023)
4. Zhou, P., Shi, W., Tian, J.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics vol. 2, pp. 207–212 (2016)
5. Schick, T., Schtze, H.: Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676 (2020)
6. Shen, S., Zhou, H., Wu, T.: Event causality identification via derivative prompt joint learning. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 2288–2299 (2022)
7. Liang, X., Zhang, N., Cheng, S, et al. Contrastive demonstration tuning for pre-trained language models. arXiv preprint arXiv:2204.04392 (2022)

8. Garcia, D.: EDF-DER, IMA-TIEM.: COATIS, an NLP system to locate expressions of actions connected by causality links. In: International Conference on Knowledge Engineering and Knowledge Management, pp. 347–352 (1997)
9. Radinsky, K., Davidocivh, S., Markovitch, S.: Learning causality for news events prediction. In: Proceedings of the 21st International Conference on World Wide Web, pp. 909–918 (2012)
10. Blanco, E., Castell, N., Moldovan, D.I.: Causal relation extraction. Lrec **66**, 74 (2008)
11. Zhao, S., Liu, T., Zhao, S.: Event causality extraction based on connectives analysis. Neurocomputing **173**, 1943–1950 (2016)
12. Minar, M.R., Naher, J.: Recent advances in deep learning: an overview. arXiv:1807.08169 (2018)
13. Ponti, E.M., Korhonen, A.: Event-related features in feedforward neural net-works contribute to identifying causal relations in discourse. In: Sentential and Discourse-level Semantics, pp. 25–30 (2017)
14. De Silva, T.N.: Causal relation identification using convolutional neural networks and knowledge based features. Int. J. Comput. Syst. Eng. **11**(6), 696–701 (2017)
15. Martínez-Cámara, E., Shwartz, V., Gurevych, I., Dagan, I.: Neural Disambiguation of Causal Lexical Markers Based on Context. In: International Conference on Computational Semantics (2017)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2019)
17. Wu, S., He, Y.: Enriching pre-trained language model with entity information for relation classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2361–2364 (2019)
18. Kyriakakis, M., Androutsopoulos, I., Saudabayev, A.: Transfer learning for causal sentence detection. In: Proceedings of the Biomedical Natural Language Processing Workshop and Shared Task, pp. 292–297 (2019)
19. Liu, J., Chen, Y., Zhao, J.: Knowledge enhanced event causality identification with mention masking generalizations. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 3608–3614 (2020)
20. Ikuta, R., Styler, W., Hamang, M.: Challenges of adding causation to richer event descriptions. In: Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pp. 12–20 (2014)
21. Jin, G., Zhou, J., Qu, W.: Exploiting rich event representation to improve event causality recognition. Intell. Autom. Soft Comput. **30**(1) (2021)
22. Liao, T., Yang, W., Zhang, S.: Event Element Recognition Based on Improved K-means Algorithm. Springer International Publishing, pp. 262–270 (2019)
23. Jin, X., Wang, X., Luo, X.: Inter sentence and implicit causality extraction from Chinese corpus. Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, Springer International Publishing, pp. 739–751, (2020)
24. Cui, S., Yan, R.: Chinese causality extraction based on softlexicon and attention mechanism. In: Proceedings of the 21st Chinese National Conference on Computational Linguistics, pp. 190–200 (2022)
25. Liao, T., Wang, X.: Event causality extraction based on joint vector and neural network. J. Anhui Univ. Sci. Technol.(Natural Science), **42**(1), 85–192 (2022)

UZNER: A Benchmark for Named Entity Recognition in Uzbek

Aizihaierjiang Yusufu[1], Liu Jiang[1], Abidan Ainiwaer[2], Chong Teng[1],
Aizierguli Yusufu[3], Fei Li[1], and Donghong Ji[1](✉)

[1] Key Laboratory of Aerospace Information Security and Trusted Computing,
Ministry of Education, School of Cyber Science and Engineering, Wuhan University,
Wuhan, China
{azhar520A,tengchong,lifei_csnlp,dhji}@whu.edu.cn
[2] School of Information Management, Wuhan University, Wuhan, China
[3] School of Computer Science and Technology, Xinjiang Normal University, Urumqi,
China
abida1020@whu.edu.cn

Abstract. Named entity recognition (NER) is a key task in natural language processing, and entity recognition can provide necessary semantic information for many downstream tasks. However, the performance of NER is often limited by the richness of language resources. For low-resource languages, NER usually performs poorly due to the lack of sufficient labeled data and pre-trained models. To address this issue, we manually constructed a large-scale, high-quality Uzbek NER corpus of Uzbek, and experimented with various NER methods. We improved state-of-the-art baseline models by introducing additional features and data translations. Data translation enables the model to learn richer syntactic structure and semantic information. Affix features provide knowledge at the morphological level and play an important role in identifying oversimplified low-frequency entity labels. Our data and models will be available to facilitate low-resource NER.

Keywords: Named Entity Recognition · Low resource · Uzbek

1 Introduction

Named Entity Recognition (NER) is a key task in natural language processing [19], and its goal is to identify entities representing names of people, places, and organizations in text. NER has wide application scenarios, such as information extraction [9], machine translation [3], question answering system [18], etc.

In recent years, NER has made great progress on high-resource languages, and many deep learning methods have achieved high accuracy [12,22,28]. However, the training of these methods relies heavily on large-scale datasets [21]. Consequently, the most significant advances in NER have been achieved in resource-rich languages such as English [20], French [25], German [4] and Chinese [8]. In

A. Yusufu and L. Jiang—Co author.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 171–183, 2023.
https://doi.org/10.1007/978-3-031-44693-1_14

contrast, in low-resource languages, the effect of NER is still poor, which limits its understanding and processing of these languages to some extent. The biggest challenge in achieving high-quality NER is usually the lack of language resources, such as manually annotated datasets and pre-trained models.

The Uzbek language studied in this paper is one of the low-resource languages. The population of this language is about 30 million, most of them are located in Uzbekistan, and the rest are scattered in Central Asian countries and Xinjiang, China, but relatively little research has been done on natural language processing for this language. The difficulty of realizing Uzbek NER lies in the limited scale of academic datasets of the language, and the lack of large-scale annotated corpus. In order to solve this problem and promote the research of Uzbek NER, we constructed a large-scale human-annotated Uzbek named entity corpus. To address the issue of entity sparsity, we reviewed the corpus and only kept sentences that contained three or more entities. It contains nearly 11,366 sentences, covering three entity types: person name, place name, and organization name.

NER can be solved by various methods, such as sequence labeling [24], span enumeration [22], hypergraph [16] and sequence-to-sequence [28] and grid tagging [12]. Because the main goal of this paper is built a Uzbek NER dataset and set up a strong baseline, we select one of the state-of-the-art (SoTA) NER model based on grid tagging as our baseline. Grounded on this model, we consider the characteristics of Uzbek and extend it by incorporating unique affix feature information of the language and expanding the training corpus by translating Cyrillic text into Latin.

Moreover, BERT [6] and BiLSTM [10] are used to provide contextualized word representations, combine them with affix feature representations to form a 2D grid of word pairs, and use multi-grained 2D convolutions to reproduce word pair representations. Finally, we employ a common predictor using dual-effect and multi-layer perceptron classifiers to generate all possible entity mentions. Our results show significant performance improvements in Uzbek NER.

In comparison to four baseline models, our proposed model [1]outperforms them by improving F1 scores by 0.34%, and the grid-tagging-based method performs better due to its attention to both entity boundary and information inside. Our model improves performance by 0.46% and 0.58% when adding affix features and augmenting the corpus with translation data, respectively.

Our contributions are as follows: 1) We constructed the first high-quality Uzbek NER corpus; 2) We introduced affix features and adopted data augmentation methods to improve the performance of NER in Uzbek; 3) Our model outperformed existing methods, achieves the state-of-the-art performance, and sets a new benchmark for the Uzbek NER task.

Our work shows that for low-resource language NER tasks, data augmentation and feature engineering are also two improvement directions. Abundant data and knowledge can help the model to learn a more generalized language representation, overcome the limitations of data scarcity, and thus greatly improve the

[1] Code is available at https://github.com/azhar520/NER.

performance of the model. This provides a strong reference for further advancing low-resource language processing.

2 Related Work

In low-resource scenarios, named entity recognition faces some challenges, such as the lack of large-scale annotation data, the quality of annotation data, and the consistency of annotation standards. In order to solve these problems, the research of low-resource entity naming recognition has emerged in recent years.

In past studies, many researchers have explored the use of cross-language transfer to solve the problem of low-resource named entity recognition. These studies show that using existing high-resource language annotated data to train a model and then transferring the model to a low-resource language can effectively improve the performance of named entity recognition in low-resource languages. For example [11] and others used the method of transfer learning to perform named entity recognition on Indonesian, and the results showed that the method performed better than the baseline model on low-resource languages. Similarly, Sun et al. (2018) [23] migrated an English entity recognition model to Hungarian and Italian, and achieved good results.

In addition to cross-language transfer, some researchers have explored human-annotated low-resource named entity recognition methods. This approach trains high-quality named entity recognition models by leveraging expert annotators to annotate a small amount of data. For example, Al-Thubaity et al. (2022) [2] used human-annotated Arabic datasets to train named entity recognition models and achieved good results. Similarly, Truong et al. (2021) [26] used a manually annotated Vietnam dataset to train an named entity recognition model and achieved higher performance than the baseline on the test set. In addition to the above methods, some researchers have explored the method of combining cross-language transfer and human annotation. This method uses cross-language transfer to leverage knowledge of high-resource languages, and then uses a small amount of human-labeled data to tune the model to achieve better results. For example, Adelani et al. (2021) [1] used cross-lingual transfer and human-annotated data to solve the problem of named entity recognition in African languages and achieved higher performance than the baseline.

Uzbek belongs to the Altaic language family and has its own grammar and rich morphological structure. Therefore, there are some special problems and challenges in the field of Uzbek NER. Although there are some researches on Uzbek natural language processing, it does not specifically involve the field of named entity and recognition. In order to fill this gap, we have done three aspects of work. First, we constructed a news-based Uzbek named entity and recognition corpus. Second, we increased the number of entity placeholders for Uzbek Cyrillic to Uzbek Latin multilingual conversion of the corpus to increase the diversity of the data set, reduce the risk of over fitting, and improve cross-language performance. Thirdly, we conducted various experiments on the corpus and incorporated affix features to provide morphological-level knowledge, with the aim of enhancing the accuracy and robustness of our entity recognition system.

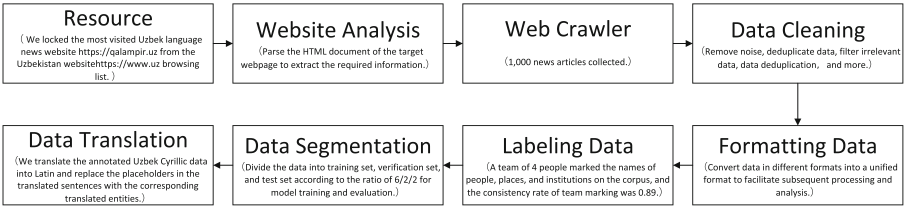

Resource
（We locked the most visited Uzbek language news website https://qalampir.uz from the Uzbekistan websitehttps://www.uz browsing list.）
Website Analysis
（Parse the HTML document of the target webpage to extract the required information.）
Web Crawler
（1,000 news articles collected.）
Data Cleaning
（Remove noise, deduplicate data, filter irrelevant data, data deduplication, and more.）
Formatting Data
（Convert data in different formats into a unified format to facilitate subsequent processing and analysis.）
Labeling Data
（A team of 4 people marked the names of people, places, and institutions on the corpus, and the consistency rate of team marking was 0.89.）
Data Segmentation
（Divide the data into training set, verification set, and test set according to the ratio of 6/2/2 for model training and evaluation.）
Data Translation
（We translate the annotated Uzbek Cyrillic data into Latin and replace the placeholders in the translated sentences with the corresponding translated entities.）

Fig. 1. Our overall workflow for dataset construction.

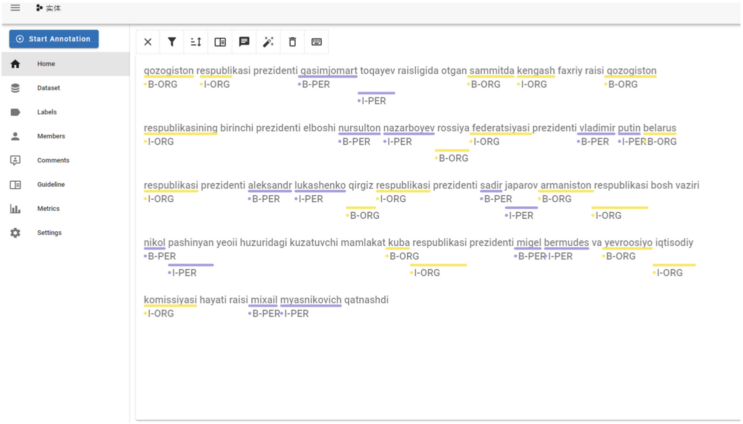
3 Dataset Construction

3.1 Data Collection and Preprocessing

First, we locked the most visited Uzbek language news website[2] from the Uzbekistan website[3] browsing list. We then analyzed the website to determine the information that we wanted to crawl, which included the title, text, time, and author of news articles. 1,000 news articles were collected by web crawler. The data was then cleaned to remove HTML tags, useless characters, and other extraneous information. This was done to ensure that the data was in a consistent format and ready for subsequent analysis. Then, the cleaned data was stored in a database or a text file to facilitate subsequent analysis and processing. Finally, we obtained the original corpus consisting of 49,019 sentences. The flowchart is shown in Fig. 1.

3.2 Data Annotation and Postprocessing

Our corpus is annotated by 4 annotators, two men and two women. They are graduate students, linguistics majors, non-native speakers but proficient in



qozogiston respublikasi prezidenti qasimjomart toqayev raisligida otgan sammitda kengash faxriy raisi qozogiston
B-ORG I-ORG B-PER I-PER B-ORG I-ORG B-ORG

respublikasining birinchi prezidenti elboshi nursulton nazarboyev rossiya federatsiyasi prezidenti vladimir putin belarus
I-ORG B-PER I-PER B-ORG I-ORG B-PER I-PER B-ORG

respublikasi prezidenti aleksandr lukashenko qirgiz respublikasi prezidenti sadir japarov armaniston respublikasi bosh vaziri
I-ORG B-PER I-PER B-ORG I-ORG B-PER I-PER B-ORG I-ORG

nikol pashinyan yeoii huzuridagi kuzatuvchi mamlakat kuba respublikasi prezidenti migel bermudes va yevroosiyo iqtisodiy
B-PER I-PER B-ORG I-ORG B-PER I-PER B-ORG I-ORG

komissiyasi hayati raisi mixail myasnikovich qatnashdi
I-ORG B-PER I-PER

Fig. 2. Annotation schema and platform.

[2] https://qalampir.uz.

[3] https://www.uz.

Uzbek. After nearly 3 months, the annotation was completed on the doccano platform. The tool supports visualization and is easy to use. A typical example is shown in Fig. 2.

Specific steps are as follows: First, we trained the annotators, informed them of the purpose of annotation, and the specific task content, and showed annotation cases for them to learn and discuss, answered their questions, and finally explained the operating specifications of the annotation system, and precautions.

Secondly, we divided the data into 4 parts, and divided the annotators into 2 groups with male and female collocations. Each person marked a piece of data. After each completed, the members of the same group exchanged data with each other for cross-labeling. Inter-annotator agreement was 0.89, as measured by span-level Cohen's Kappa [5]. We organize four people to discuss the inconsistency and until a consensus is reached.

Finally, we traversed all the data to check and corrected a few errors. The annotator training process took about a week. After annotator training, we provide each annotator with a batch for a period of one month. Then, we asked them to exchange data with a team member to start a new round of annotations for a month, unless they were tired, bored, or sick. We do this to ensure the quality of annotations. They were checked for consistency after two months, inconsistencies were dealt with together, and then another review was conducted, and thus, after nearly three months, we finally obtained a golden corpus with 49,019 labels consisting of 879,822 tokens, including 24,724 names of people, 35,743 names of places, and 25,697 names of institutions.

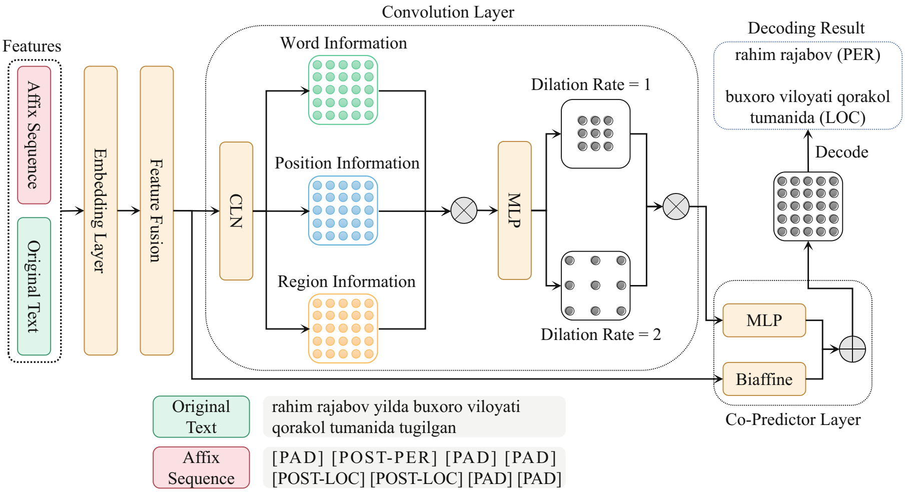


Fig. 3. Our model architecture. $\boldsymbol{H}^x$ and $\boldsymbol{H}^c$ represent the original text embedding and the affix sequence embedding respectively. $\oplus$ and $\otimes$ represent element-wise addition and concatenation operations. Both the convolutional layer and the collaborative prediction layer come from the SoTA model [12].

Due to the sparsity of entity data in news sentences, we retain sentences with three or more entities in a sentence. After screening, we got a corpus of 11,366 sentences consisting of 402,707 tokens. The longest sentence has 56 words, the shortest sentence has only 5 words, and the average length is 35.4 tokens. The corpus contains a total of 78,385 named entities. Among them, the longest name of a person is composed of 5 tokens, the name of a place is composed of 4 tokens, and the name of an organization is composed of 14 tokens. We randomly divide the gold-marked corpus into training sets/verification set/testing set, the ratio is 6/2/2, and the statistical results of the corpus are shown in the Table 1.

Table 1. Dataset statistics

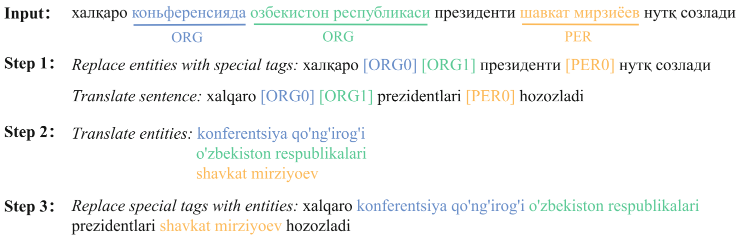
Data	**Sentence**	**Token**	**Token/Sent**	**PER**	**LOC**	**ORG**	**Discontinuous**
Train	7,366	311,885	42.35	14,137	11,029	12,084	4.26%
Dev	2,000	45,035	22.50	3,094	3,974	3,142	3.16%
Test	2,000	45,787	22.90	4,252	3,199	3,054	3.66%

4 Method

Our model is an improvement on W^{2}NER [12], including embedding layer, convolution layer and co-predictor layer. The difference is that the affix feature is integrated into the model. Our model architecture is shown in Fig. 3.

4.1 Features

Before introducing the model, we briefly introduce the construction of the affix sequence. Uzbek language contains many affix features, which are helpful for identifying entities. Therefore, we count the corresponding affixes according to the type. During the labeling process, we found that Uzbek personal names usually end with “ov”, “ova”, and “lar”; place names usually end with “stan”, “shahr”, “ko’li”, etc.; institution names often end with “markaz”, “kompaniya”, “tashkilot” and other affix endings. These affixes include 64 place name prefixes, 23 personal name prefixes, 24 personal name suffixes, and 105 organizational name suffixes. Based on our statistics, we use four special tags to represent affix features, namely *[PRE-PER]* (PER class prefix), *[POST-PER]* (*PER* class postfix), *[POST-LOC]* (LOC class postfix) and *[POST-ORG]* (ORG class postfix). If there is no affix in the token, it will be filled with *[PAD]*. In this way, we can construct the affix sequence corresponding to the original text, such as the example shown in the bottom part of Fig. 3, an original text and its corresponding affix sequence.


Input: халқаро коньференсияда озбекистон республикаси президенти шавкат мирзиёев нутқ созлади

Step 1: *Replace entities with special tags:* халқаро [ORG0] [ORG1] президенти [PER0] нутқ созлади

Translate sentence: xalqaro [ORG0] [ORG1] prezidentlari [PER0] hozozladi

Step 2: *Translate entities:* konferentsiya qo'ng'irog'i
o'zbekiston respublikalari
shavkat mirziyoev

Step 3: *Replace special tags with entities:* xalqaro konferentsiya qo'ng'irog'i o'zbekiston respublikalari prezidentlari shavkat mirziyoev hozozladi

Fig. 4. Flowchart for data translation. Input is Cyrillic, and finally translated into Latin.

4.2 Data Translation

Since Uzbek includes Latin and Cyrillic, inspired by Liu et al. [14], in the data preprocessing stage, we consider translating Cyrillic to Latin to augment the training corpus. The specific translation process is divided into three steps: first, replace the entities in the Cyrillic sentence with special tags, and then translate into Latin; then translate the Cyrillic entities into Latin one by one; finally fill in the Latin entities into the translated Latin sentence. The whole process is translated using the Google translation model, and the overall flow chart is shown in Fig. 4.

4.3 NER Model

Embedding Layer. The embedding layer is the same as W^{2}NER, including BERT [6] and BiLSTM [10], but the input not only has the original text $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$ of length n, but also the affix sequence $C = \{c_1, c_2, \ldots, c_n\} \in \mathbb{R}^n$. After the embedding layer, the original text embedding $\boldsymbol{H}^x$ and the affix sequence embedding $\boldsymbol{H}^c$ are obtained:

$$\begin{aligned} \boldsymbol{H}^x &= \{\boldsymbol{h}_1^x, \boldsymbol{h}_2^x, \ldots, \boldsymbol{h}_n^x\} \in \mathbb{R}^{n \times d_h}, \\ \boldsymbol{H}^c &= \{\boldsymbol{h}_1^c, \boldsymbol{h}_2^c, \ldots, \boldsymbol{h}_n^c\} \in \mathbb{R}^{n \times d_h}, \end{aligned} \tag{1}$$

where $\boldsymbol{h}_i^x$, $\boldsymbol{h}_i^c \in \mathbb{R}^{d_h}$ are the representations of the i-th token, and d_h represents the dimension of a token representation.

After that, we sum $\boldsymbol{H}^x$ and $\boldsymbol{H}^c$ at the element level to get the text embedding $\boldsymbol{H}^s = \{\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, \ldots, \boldsymbol{h}_n^s\} \in \mathbb{R}^{n \times d_h}$ that incorporates affix features. The subsequent process is the same as W^{2}NER, so we will only briefly introduce it.

Convolution Layer. After obtaining the text embedding $\boldsymbol{H}^s$ that incorporates the affix feature, the Conditional Layer Normalization (CLN) mechanism is used to generate a 2D grid $\boldsymbol{V}$, where each item $\boldsymbol{V}_{ij}$ in $\boldsymbol{V}$ is a representation of a word pair (x_i, x_j), so:

$$\boldsymbol{V}_{ij} = \mathrm{CLN}(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s) = \gamma_{ij} \odot (\frac{\boldsymbol{h}_j^s - \mu}{\sigma}) + \lambda_{ij}, \tag{2}$$

where $\boldsymbol{h}_i$ is the condition to generate the gain parameter $\gamma_{ij} = \boldsymbol{W}_\alpha \boldsymbol{h}_i^s + b_\alpha$ and bias $\lambda_{ij} = \boldsymbol{W}_\beta \boldsymbol{h}_i^s + b_\beta$ of layer normalization. $\boldsymbol{W}_\alpha$, $\boldsymbol{W}_\beta \in \mathbb{R}^{d_h \times d_h}$ and $\boldsymbol{b}_\alpha$, $\boldsymbol{b}_\beta \in \mathbb{R}^{d_h}$ are trainable weights and biases respectively. μ and σ are the mean and standard deviation across the elements of $\boldsymbol{h}_j^s$.

Then word, position and sentence information on the grid is modeled, where $\boldsymbol{V} \in \mathbb{R}^{n \times n \times d_h}$ represents word information, $\boldsymbol{V}^p \in \mathbb{R}^{n \times n \times d_{h_p}}$ represents the relative position information between each pair of words, and $\boldsymbol{V}^r \in \mathbb{R}^{n \times n \times d_{h_r}}$ represents the region information for distinguishing lower and upper triangle regions in the grid. They are concatenated them to get the position-region aware representation of the grid:

$$\boldsymbol{Z} = \mathrm{MLP}_1([\boldsymbol{V}; \boldsymbol{V}^p; \boldsymbol{V}^r]) \in \mathbb{R}^{n \times n \times d_{h_z}}, \tag{3}$$

Finally, the multiple 2D dilated convolutions (DConv) with different dilation rates are used to capture the interactions between the words with different distances, formulated as:

$$\boldsymbol{Q} = \mathrm{GeLU}(\mathrm{DConv}(\boldsymbol{Z})), \tag{4}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N \times N \times d_q}$ is the output and GeLU is a activation function.

Co-Predictor Module. Finally, the word pair relationship is predicted by the co-predictor, which includes the MLP predictor and the biaffine predictor. Therefore, we take these two predictors to calculate the two independent relationship distributions (x_i, x_j) of word pairs at the same time, and combine them as the final prediction. For MLP, the relationship score of each word pair (x_i, x_j) is calculated as:

$$\boldsymbol{y}_{ij}^{'} = \mathrm{MLP}_2(\boldsymbol{Q}_{ij}), \tag{5}$$

The input of the biaffine predictor is the input $\boldsymbol{H}^s$ of the CLN, which can be considered as a residual connection. Two MLPs are used to calculate the representation of each word in the word pair (x_i, x_j). Then, the relationship score between word pairs (x_i, x_j) is calculated using a biaffine classifier:

$$\boldsymbol{y}_{ij}^{''} = \boldsymbol{s}_i^\top \boldsymbol{U} \boldsymbol{o}_j + \boldsymbol{W}[\boldsymbol{s}_i; \boldsymbol{o}_j] + \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters, and $\boldsymbol{s}_i = \mathrm{MLP}_3(\boldsymbol{h}_i^s)$ and $\boldsymbol{o}_j = \mathrm{MLP}_4(\boldsymbol{h}_j^o)$ represent the subject and object representations respectively. Finally, we combine the scores from the MLP and biaffine predictors to get the final score:

$$\boldsymbol{y}_{ij} = \mathrm{Softmax}(\boldsymbol{y}_{ij}^{'} + \boldsymbol{y}_{ij}^{''}). \tag{7}$$

Decoding Algorithm. We decode entities based on two designed word pair relationships, which are (1) Next-Neighboring-Word (NNW) indicates that the word pair (x_i, x_j) belongs to an entity, and the next word of x_i in the entity is x_j. (2) Tail-Head-Word-* (THW-*) indicates that the word in the row of the grid is the tail of the entity, and the word in the column of the grid is the head of the entity. $*$ indicates the entity type.

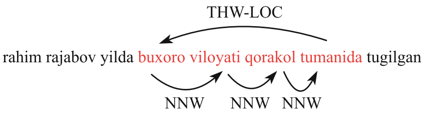

THW-LOC
rahim rajabov yilda buxoro viloyati qorakol tumanida tugilgan
NNW
NNW
NNW

Fig. 5. An example showing the process of identifying entities.

We also provided an example in Fig. 5 to explain the process of identifying different types of entities. For example, for the PER entity "oydin", it can be known from the THW-PER relationship that "oydin" is both the head and the tail of an entity, so it itself is an entity with a length of 1 and its category is PER. Then, for the ORG entity "ozbekiston respublikasi oliy majlis", by using the NNW relationship with the subject "ozbekiston" and object "respublikasi", we recognize "ozbekiston respublikasi" as a part of the entity. Similarly, "respublikasi oliy" and "oliy majlis" is also recognized in the same way. Then, by using the THW-ORG, we recognize "ozbekiston" and "majlis" are the head and tail of the entity, so that "ozbekiston respublikasi oliy majlis" can be recognized completely and its category is ORG.

5 Experiments

5.1 Experimental Setting

We conduct experiments on our UzNER (Latin) dataset to evaluate the effectiveness of our proposed model. If the token sequence and type of a predicted entity are exactly the same as those of a gold entity, the predicted entity is regarded as true-positive. We run each experiment three times and report their average value.

Our model uses *bert-base-multilingual-cased* [6] as the backbone network. We set a dropout of 0.5 on both the output representations of the BERT and convolutional module, and a dropout of 0.33 on the output representations of the co-predictor module, the learning rate of BERT and the learning rate of other modules are 1e-5 and 1e-3 respectively, the batch size is 12, and d_q can choose 64, 80, 96 and 128. The hyper-parameters are adjusted according to the fine-tuning on the development sets.

5.2 Baselines

We use some existing models of different methods as our baseline models. All baseline models are trained using an expanded corpus after translation. In addition, since our corpus is in Uzbek, the backbone network uses multilingual pre trained models.

BiLSTM+CRF [10] is the most basic sequence labeling model. Due to the presence of discontinuous entities in the dataset, we use BIOHD [24] tags to decode the entities. **BartNER** [28] is based on the Seq2Seq method, and they use pre-trained language models to solve NER tasks. We use *mbart-large-cc25* [15] as the backbone network. **W^{2}NER** [12] is based on a grid labeling method, which identifies all possible entities through word pair relationships. We use *bert-base-multilingual-cased* [6] as the backbone network. **UIE** [17] is a unified text-to-structure generation framework. UIE is not pre-trained in our experiments. We use *mt5-base* [27] as the backbone network.

5.3 Comparison with Baselines

The comparison results with the baseline models are shown in Table 2. We have the following findings: 1) Our model outperforms four baseline models. Compared with the method of Li et al. (2022) [12], our method improves the F1s by 0.34; 2) The grid-tagging-based method outperforms other methods, because the method not only pays attention to the boundary of the entity, but also pays attention to the information inside the entity. 3) The effect of BiLSTM+CRF is the worst, which is natural, because its structure is too simple compared to other models, and it can learn too little knowledge.

Table 2. Comparison with baseline models and ablation experiments. Green scores represent the best result in that column, blue scores represent the second best result in that column excluding ablation results.

	P	**R**	**F1**
BiLSTM+CRF [10]	86.81	79.28	82.87
w/o Data Translation	84.62	80.05	82.27
BartNER [28]	92.34	90.16	91.23
w/o Data Translation	90.42	88.09	89.24
W^{2}NER [12]	92.47	90.29	91.37
w/o Data Translation	92.35	90.01	91.16
UIE [17]	91.28	87.13	89.16
w/o Data Translation	87.23	83.41	85.28
Ours	**92.83**	**90.63**	**91.71**
w/o Affix	92.39	90.13	91.25
w/o Data Translation	91.93	90.34	91.13

5.4 Ablation Studies

The results of the ablation experiments are shown in Table 2. We mainly analyzed the two improvement schemes that we proposed. First, when we remove the affix features, the performance of our model on F1s drops by 0.46, indicating that the fusion of affix features is effective and can better improve the performance of the model. We then train the model without translation data, and our model performance drops by 0.58 on F1s, which is natural, more data allows the model to learn more features. In addition, for the baseline model, we also train on the augmented data without translation, and the performance is also reduced.

Table 3. Performance comparison among different entity classes. Green scores and blue scores represent the best and second best results in that column excluding ablation results.

	LOC			PER			ORG		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BiLSTM+CRF [10]	90.77	87.42	89.06	86.30	73.73	79.52	82.55	78.80	80.63
w/o Data Translation	89.03	87.09	88.05	85.70	76.46	80.82	78.39	78.28	78.33
BartNER [28]	94.42	**95.79**	95.10	**91.32**	85.37	**88.24**	89.80	91.88	90.83
w/o Data Translation	93.22	95.08	94.14	88.84	82.77	85.70	87.82	89.54	88.67
W^{2}NER [12]	95.85	94.05	94.94	90.02	85.74	87.83	91.80	92.23	92.01
w/o Data Translation	95.73	93.92	94.81	90.28	85.60	87.87	91.64	92.09	91.86
UIE [17]	94.85	90.62	92.69	91.06	81.70	86.13	91.06	89.08	90.06
w/o Data Translation	91.47	87.06	89.21	86.96	75.95	81.08	86.80	85.79	86.29
Ours	**96.41**	94.42	**95.41**	90.08	**86.45**	88.22	**92.83**	**92.52**	**92.67**
w/o Affix	95.92	94.28	95.09	89.77	85.47	87.57	92.26	92.33	92.29
w/o Data Translation	95.39	94.06	94.72	89.63	86.33	87.95	91.45	92.07	91.76

5.5 Performance Analysis on Different Entity Types

We also explored the effectiveness of our method and baseline method on three entity classes. The comparison results with the baseline model are shown in Table 3. First, our method outperforms all baseline models on LOC and ORG by 0.31 and 0.66 on F1s compared to the second-best results. The performance achieved on F1s is the second best compared to the baseline model, only 0.02 lower than the best performance.

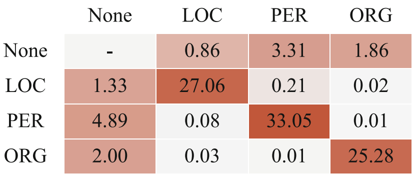
Table 4. Error analysis experiment. EBE and ETE represent Entity Boundary Error and Entity Type Error, respectively.

Error Type	EBE	ETE
All (%)	99.65	0.35
LOC (%)	22.29	0.11
PER (%)	47.48	0.21
ORG (%)	29.87	0.03

In the lower part of Table 3, we also analyze the ablation results on different entity classes. First, with the removal of affix features, the performance of our model on all three types of entities degrades. Then, without training the model with translated data, the performance of our model drops on all three types of entities. Finally, the performance of the baseline model on all three types of entities also decreases when trained without translation data.

5.6 Error Analysis

We also performed error analysis to learn more about our model. The results are shown in the of Table 4. Most of the errors come from boundary errors, accounting for 99.65% of all errors, because entity boundaries are difficult to identify, which is a well-known problem in previous work [7,13]. In addition, we also analyzed the proportion of different types of errors. Regardless of the type of error, the PER entity has the largest proportion of errors. This is because PER has higher text diversity and the model is more difficult to predict more PER entities. Finally, Fig. 6 is a heat map of the confusion matrix of error analysis. The diagonal line represents the proportion of correct recognition, so it is the highest proportion, which is natural. In addition, the proportion of the first row and the first column is next, which is reasonable, because the proportion of these two parts is equivalent to the boundary error, which is consistent with the results in Table 4.



	None	LOC	PER	ORG
None	-	0.86	3.31	1.86
LOC	1.33	27.06	0.21	0.02
PER	4.89	0.08	33.05	0.01
ORG	2.00	0.03	0.01	25.28

Fig. 6. The confusion matrix for error analysis. None represents non-entity. Numbers represent percentages. Rows and columns represent the gold and predicted results, respectively.

6 Conclusion

Our study proposes a novel approach to enhance the state-of-the-art model for Uzbek NER by incorporating unique affix feature information of the language and expanding the training corpus by translating Cyrillic text into Latin. Our proposed model outperforms four baseline models with a significant F1 score improvement of 0.34%, demonstrating the effectiveness of our approach. The grid-tagging-based method is found to be superior to other methods due to its attention to both entity boundary and information inside. Our findings highlight the importance of incorporating unique language features and utilizing advanced neural network architectures for NER tasks. In the future, further exploration of other language-specific features and integration of cross-lingual transfer learning can potentially improve the performance of NER models for low-resource languages like Uzbek.

Acknowledgment. This work is supported by the Natural Science Program of Xinjiang Uygur Autonomous Region for the Construction of Innovation Environment (Talents and Bases) (Special Training of Scientific and Technological Talents of Ethnic Minorities)(2022D03001), the National Natural Science Foundation of China (No. 62176187; No. 61662081), the Major Projects of the National Social Science Foundation of China (No.11&ZD189; No.14AZD11), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015).

References

1. Adelani, D.I., et al.: Masakhaner: named entity recognition for African languages. Trans. Assoc. Comput. Linguist. **9**, 1116–1131 (2021)
2. Al-Thubaity, A., Alkhereyf, S., Alzahrani, W., Bahanshal, A.: Caraner: the Covid-19 Arabic named entity corpus. In: WANLP@EMNLP 2022, pp. 1–10 (2022)
3. Balabantaray, R.: Name entity recognition in machine translation. Emerg. Technol **1**(3), 3 (2010)
4. Benikova, D., Biemann, C., Reznicek, M.: Nosta-d named entity annotation for German: Guidelines and dataset. In: LREC 2014, pp. 2524–2531 (2014)
5. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019, pp. 4171–4186 (2019)
7. Fei, H., Ji, D., Li, B., Liu, Y., Ren, Y., Li, F.: Rethinking boundaries: end-to-end recognition of discontinuous mentions with pointer networks. In: AAAI 2021, pp. 12785–12793 (2021)
8. Ji, B., et al.: A hybrid approach for named entity recognition in Chinese electronic medical record. BMC Med. Inform. Decis. Mak. **19**(2), 149–158 (2019)
9. Krallinger, M., Valencia, A.: Text-mining and information-retrieval services for molecular biology. Genome Biol. **6**(7), 1–8 (2005)
10. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: HLT-NAACL 2016, pp. 260–270 (2016)
11. Leonandya, R., Ikhwantri, F.: Pretrained language model transfer on neural named entity recognition in indonesian conversational texts. arXiv preprint arXiv:1902.07938 (2019)

12. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: AAAI 2022. vol. 36, pp. 10965–10973 (2022)
13. Liu, J., et al.: TOE: a grid-tagging discontinuous NER model enhanced by embedding tag/word relations and more fine-grained tags. IEEE/ACM Trans. Audio, Speech, Lang. Process. **31**, 177–187 (2022)
14. Liu, L., Ding, B., Bing, L., Joty, S., Si, L., Miao, C.: Mulda: a multilingual data augmentation framework for low-resource cross-lingual NER. In: ACL/IJCNLP 2021, pp. 5834–5846 (2021)
15. Liu, Y., et al.: Multilingual denoising pre-training for neural machine translation. Trans. Assoc. Comput. Linguist. **8**, 726–742 (2020)
16. Lu, W., Roth, D.: Joint mention extraction and classification with mention hypergraphs. In: EMNLP 2015, pp. 857–867 (2015)
17. Lu, Y., et al.: Unified structure generation for universal information extraction. In: ACL 2022, pp. 5755–5772 (2022)
18. Mollá, D., Van Zaanen, M., Smith, D.: Named entity recognition for question answering. In: ALTA 2006, pp. 51–58 (2006)
19. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
20. Ringland, N., Dai, X., Hachey, B., Karimi, S., Paris, C., Curran, J.R.: NNE: a dataset for nested named entity recognition in english newswire. arXiv preprint arXiv:1906.01359 (2019)
21. Rosenfeld, J.S.: Scaling laws for deep learning. arXiv preprint arXiv:2108.07686 (2021)
22. Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., Lu, W.: Locate and label: a two-stage identifier for nested named entity recognition. In: ACL/IJCNLP 2021, pp. 2782–2794 (2021)
23. Sun, P., Yang, X., Zhao, X., Wang, Z.: An overview of named entity recognition. In: IALP 2018, pp. 273–278. IEEE (2018)
24. Tang, B., Hu, J., Wang, X., Chen, Q.: Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF. In: Proceedings of the Wireless Communications and Mobile Computing 2018 (2018)
25. Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., Navigli, R.: Wikineural: combined neural and knowledge-based silver data creation for multilingual NER. In: EMNLP (Findings) 2021, pp. 2521–2533 (2021)
26. Truong, T.H., Dao, M.H., Nguyen, D.Q.: Covid-19 named entity recognition for Vietnamese. arXiv preprint arXiv:2104.03879 (2021)
27. Xue, L., et al.: mt5: A massively multilingual pre-trained text-to-text transformer. In: NAACL-HLT 2021, pp. 483–498 (2021)
28. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various NER subtasks. In: ACL/IJCNLP 2021, pp. 5808–5822 (2021)

Evaluation Framework for Poisoning Attacks on Knowledge Graph Embeddings

Dong Zhu[1], Yao Lin[1], Le Wang[1,4](✉), Yushun Xie[2], Jie Jiang[1], and Zhaoquan Gu[3,4]

[1] Guangzhou University, Guangzhou, China
{2112106282,2112233084,2112011051}@e.gzhu.edu.cn
[2] University of Electronic Science and Technology of China, Chengdu, China
yshxie@std.uestc.edu.cn
[3] Harbin Institute of Technology (Shenzhen), Shenzhen, China
guzhaoquan@hit.edu.cn
[4] Peng Cheng Laboratory, Shenzhen, China
wanglelemail@yeah.net

Abstract. In the area of knowledge graph embedding data poisoning, attackers are beginning to consider the importance of poisoning sample exposure risk while increasing the toxicity of the poisoning sample. On the other hand, we have found that some researchers incorrectly assess the effectiveness of poisoning attacks without considering the impact of the data-adding operation on model performance. Also, there is currently no definition of the Stealthiness of poisoning attacks. To address this issue, we provide an objective and unified framework for evaluating complex and diverse poisoning strategies. We design a controlled experiment on poisoning attacks to obtain objectively correct poisoning effects, and propose toxicity Dt to evaluate the poisoning performance of poisoning attacks and stealthiness Ds to evaluate the exposure risk of poisoning attacks. In designing the metrics, we fully considered the performance of the control model and the generalizability of the attacked model so that the data poisoning effect can be objectively and correctly evaluated. We compared 12 recently proposed KG attack methods on two different benchmark datasets to verify the objectivity and correctness of our evaluation criteria and to analyze the impact of the generalizability of the attacked model.

Keywords: Knowledge graph embedding · Data poisoning attack · Attack stealthiness · Evaluation framework

1 Introduction

The field of knowledge graphs holds significant importance in computer science research and finds extensive applications in industrial tasks such as information extraction [8], dialogue systems [15], and recommendation systems [13]. However, knowledge graphs often suffer from incompleteness, necessitating completion techniques to enhance their

This work is supported in part by the Major Key Project of PCL (Grant No. PCL2022A03), Guangdong High-level University Foundation Program (SL2022A03J00918) and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 184–196, 2023.
https://doi.org/10.1007/978-3-031-44693-1_15

accuracy. One such approach is knowledge graph embedding, which involves representing graph entities and relations in a continuous vector space. Prominent embedding models, such as TransE [4], DistMult [12], and ConvE [6] models, which all exhibit good expressive power. However, these models are vulnerable to poisoning attacks [9], which can degrade their performance.

In previous studies [3] on poisoning attacks, researchers expanded the training set of a model by introducing adversarial data, thereby generating a new model. The effectiveness of the poisoning attack was assessed by comparing the performance of the target samples between the two models. However, this approach disregarded the influence of the data augmentation process itself, including its impact on the model's performance as well as the target samples.In the realm of defending against poisoning attacks, collective robustness certification [5] has emerged as an innovative and efficacious approach. This methodology employs hash-based deterministic sampling as a potent defense mechanism to detect attacks by constraining the influence sphere of each poisoned sample. The effectiveness of this defensive strategy is derived from the fact that a majority of assailants overlook the range of influence and exposure risk associated with the poisoned samples, as depicted in Fig. 1. To assess the extent of interference on the target samples and its impact on other non-target samples, we propose an evaluative framework that accounts for the inherent influence of data augmentation on the model's performance. This framework facilitates an objective and accurate appraisal of the poisoning effects on target samples and the concealment of non-target samples.

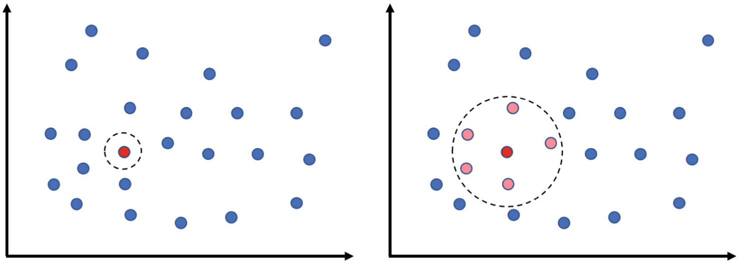


Fig. 1. Comparison between stealthy and exposed attacks. Exposed attacks affect the surrounding entities of the target node making the attacks easy to be detected.

During this process, the inclusion of poisoned samples can have an influence on the model's capacity for generalization, consequently impacting the assessment of attack effectiveness. To address this, we propose the introduction of a generalization parameter for calibration.

We are the first to make a unified evaluation of poisoning effects. By establishing a control model that closely resembles the poisoned model, as illustrated in Fig. 2, each assessment of attack effectiveness is based on a controlled experiment rather than the original model. We propose specific metrics for poisoning and concealment, which respectively measure the attack capability on target samples and the level of concealment. Furthermore, we introduce a comprehensive scoring system to evaluate the overall effectiveness of poisoning attacks. Detailed information can be found in Sect. 4.Our contributions are summarized below:

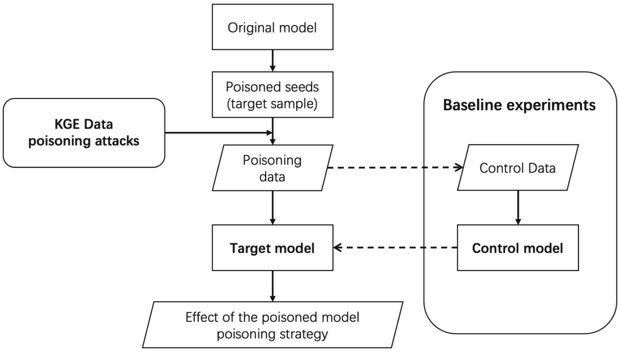

Original model
Poisoned seeds (target sample)
KGE Data poisoning attacks
Poisoning data
Target model
Effect of the poisoned model poisoning strategy
Baseline experiments
Control Data
Control model

Fig. 2. The attacker's data poisoning process, and the dotted line shows the control experiment we added.

1) Previous studies overlooked the impact of the data augmentation process on model performance. We conducted experiments to analyze this aspect and subsequently revised the evaluation of poisoning effectiveness.
2) We elucidated the influence of poisoning attacks on model generalization and introduced the concept of the generation rate of poisoned data to assess its impact on attack effectiveness.
3) We proposed an evaluation framework specifically designed for poisoning attacks on knowledge graph embeddings. This framework encompasses the calculation and experimental process of poisoning and stealthiness, offering an objective measure of poisoning attack performance.
4) To validate the effectiveness of the evaluation framework, we conducted extensive experiments using two widely-used benchmark datasets. Four models and twelve attack methods were employed, allowing us to assess the accuracy of the framework and its capability to evaluate the performance of poisoning attacks.

2 Related Work

2.1 Knowledge Graph Embedding Poisoning Attacks and Defenses

Knowledge Graph Embedding. The modeling of relationships in translation models for knowledge graph embeddings involved representing them as translation vectors from the head entity to the tail entity, as exemplified by TransE [4]. Semantic matching models utilized similarity-based scoring functions to assess the plausibility of facts by matching the latent semantics of entities and relationships reflected in vector space representations. Examples of such models include DistMult [12] and ComplEx [11]. Neural network models capitalized on their powerful learning capabilities to capture feature representations of entities and relationships include ConvE [6].

Data Poisoning Attacks. In knowledge graph embedding data poisoning attacks, attackers can submit poisoned data through crowdfunding to complete the attack [7,10]. Typically, attackers first select target samples and use them as poisoning seeds to generate poisoning data that can affect the embeddings of the target samples, and then add them to the training data.

The simplest and most effective method is random attack, which can be divided into global random attack and local random attack. The former generates poisoned samples randomly distributed throughout the entire knowledge graph, while the latter generates random samples distributed around the target sample based on the target sample. Zhang [14] proposed two attack methods, adding attack and deleting attack, for knowledge graph embedding data poisoning attacks. They only need to add poisoned samples in the direction of the target sample's gradient descent or delete the neighboring samples with the largest gradient impact on the target sample. Bhardwaj [3] started from the relationship patterns of triples, most triples have logical relationships, such as symmetric and composite. they can interfere with the results of knowledge graph embedding by disrupting the logic of reasoning. The work [2] proposed a method to calculate the contribution of the target triple's embedding and affect the target triple's completion by deleting or replacing the surrounding triples with the highest contribution. Banerjee [1] was the first to consider the stealthiness of data poisoning. They constructed a non-existent triple sequence through a sequence iterator to replace the target triple, and calculated the exposure risk of the entire sequence at each iteration. However, the authors did not propose a definition and objective calculation method for evaluating the stealthiness of attacks.

2.2 Evaluation Methods

As there is currently no standardized protocol for evaluating the effectiveness of data poisoning, this section will primarily focus on how past attackers determined the effectiveness of their poisoning attacks.

The task of knowledge graph embedding (KGE) is to embed entities E and relations R into a vector space to represent the semantic and structural information of the knowledge graph $G = \{E, R, T\}$, where $T = \{(h, r, t)h, \mid t \in E, r \in R\}$is a set of triplets (facts). During training, the best embedding is obtained by assigning higher scores to more effective triplets. To predict the accuracy of the entire knowledge graph embedding $T = \{(h, r, t)\}$, the KGE model scores all triplets in the set $T' = \{(h, r, t') \mid t' \in E\}$ and ranks them accordingly. The ranking N of the test triplet(h, r, t) is determined by its score relative to other triplets in the list. A higher ranking indicates a better embedding effect for the triplet, which is the link prediction task. The scores of link prediction for all triplets can reflect the ability of the model to preserve the semantic and topological structure when embedding the knowledge graph into the vector space.

In order to evaluate the embedding performance of the model, current work [4] observes the average ranking of all triplets in the knowledge graph with respect to link prediction. $MR = \frac{1}{|T|} \sum_{i=1}^{|T|} rank_i$(where $|T|$ is the number of triplets in the test set) or the reciprocal of the average ranking $MRR = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{rank_i}$, also known as the mean reciprocal rank.$Hits@n = \frac{1}{|T|} \sum_{i=1}^{|T|} (IIrank_i \leq n)$,(where II is the indicator function, which is 1 if the condition is true, and 0 otherwise). This metric is a common embedding evaluation metric and indicates whether the correct result is among the top N rankings in the predicted results of the triplet. If it is, it counts as 1; otherwise, it counts as 0. The sum of all counts for the test triplets divided by the number of test triplets is the value of $Hits@n$.

3 Evaluation Framework for KGE Attack

In this section, we will first present our problem formulation in Sect. 3.1, followed by a description of our experimental framework design in Sect. 3.2. Then, we will analyze how we can accurately evaluate the toxicity, stealthiness, and comprehensiveness of data poisoning attacks using the experimental data in Sect. 3.3.

3.1 Problem Framework

A knowledge graph G represents a set of facts F as a non-weighted directed graph. Each vertex represents an entity, and a directed edge between two vertices represents a relationship between two entities. We refer to a fact as a triplet $f = (h, r, t)$, where h and t are the head and tail entities, and r is the relationship from h to t. Let E be the set of all entities, R be the set of all relationships, and $P = E \times R \times E$ be the set of all possible triplets. Every fact in F is a triplet in P, so $F \subseteq P$.

An unknown subset $F' : F \subset F' \subset P$ represents all true facts, but not all facts exist in the knowledge graph. A base model is trained on all facts in F, then used to identify if a missing fact $f \in P \backslash F$ belongs to F'. Based on this, f may be added to KG to complete it. Specifically, the base model first initializes entity and relationship vectors randomly. Then a scoring function is defined to compute the confidence score of a triplet. Entity and relationship vectors are trained by maximizing the confidence score, and these embeddings predict the confidence score of each triplet $f \in P$. A higher (lower) confidence score means that a triplet is more likely to be true (conversely, less likely). Only those triplets in $P \backslash F$ with high confidence scores will be considered as facts and added to F.

A data poisoning attack targets a fact that does not belong to the existing set of facts F. Let $f^* \in F' \backslash F$ represent the target fact to be attacked. Let D represent the set of perturbations constructed based on f^*, but $f \in P \backslash F'$. $F \cup D$ represents the set of perturbed facts generated by perturbing F using D. T represents the set of same-size facts selected from $f \in F' \backslash F$ as D. $F \cup S$ represents the set of facts added to F based on S. M represents the embedding performance of the base model, $M_{F \cup D}$ represents the embedding performance of the base model trained on perturbed facts, and $M_{F \cup S}$ represents the embedding performance of the base model trained on the set of facts $F \cup T$. Compared to the embedding performance M of the base model, a good data poisoning attack will result in a significant decrease in $M_{F \cup D}$, while $M_{F \cup S}$ is almost the same as M regardless of the type of data poisoning attack chosen, because S is a true fact in $f \in F'$. If there is a difference between $M_{F \cup S}$ and M, it is due to the base model itself and not the perturbation set D.

In previous work, researchers did not distinguish between $M_{F \cup S}$ and M when evaluating the effect of data poisoning attacks. This led to a widespread overestimation of the effectiveness of data poisoning attacks, as discussed in Sect. 4.2. Additionally, attackers only care about the credibility score reduction of the target fact f^* after adding the perturbation set D, and do not consider the impact on the credibility score of facts $f \in F \cap F' \backslash D$. This neglect increases the exposure risk of data poisoning attacks. Next, we provide a formal description of the problem that our evaluation protocol addresses.

Problem Description: Given a knowledge graph G with a fixed set of facts F, the performance evaluation results M obtained from the original embedding model, and the data poisoning attack P targeting the facts $f^* \in F' \setminus F$, the objective is to accurately assess the poisoning effect of P on poisoning based on $F \cup S$ and the exposure risk based on $f \in F \cap F' \setminus D$. Additionally, the evaluation considers the influence of the model's generalization ability on the assessment.

3.2 Experimental Procedure

In this section, we begin by providing formal definitions for toxicity and stealthiness. Next, we outline the primary objectives of both the attack and control experiments. Finally, we present the calculation methods for quantifying toxicity and stealthiness, along with the combined effect indicators.

Definition 1 (Toxicity): Given a knowledge graph $F' = (h, r, t)$, the attacker selects target triples $f = (h_{\text{tar}}, r_{\text{tar}}, t_{\text{tar}})$ (where $N_{\text{tar}} \in F' \backslash F$) and generates a batch of poisoned triples f^* using a data poisoning attack P. These poisoned triples are generated based on the target triples and added to the original model's training set $T^{\text{ori}}_{\text{train}}$, resulting in a new training set $T^{\text{att}}_{\text{train}}$. Similarly, an equal number of control triples from $F \cap F' \setminus D$ are added to the training set $T^{\text{ori}}_{\text{train}}$ of the original model to create a new training set $T^{\text{con}}_{\text{train}}$. After updating the model, the MRR of the target triples is measured in the attack experiment and compared to that in the control experiment. The decrease in MRR of the target triple in the attack experiment, relative to the control experiment, is used to quantify the toxicity.

Definition 2 (Stealthiness): In a given knowledge graph $F' = \{(h, r, t)\}$, after the model is updated, the degree to which the added poisoned triples interfere with the set of triples that were not selected by the attacker N_{con} ($N_{\text{con}} \in F \cap F' \backslash D$, $|N_{\text{con}}| = |N_{\text{tar}}|$ with the same distribution) is referred to as stealthiness.

Attack Experiment. In the attack experiment, the attacker selects a target triplet N_{tar} and generates poisoned samples P using different attack strategies. The resulting poisoned samples are added to the original training set $T^{\text{ori}}_{\text{train}}$ to create a new training set $T^{\text{att}}_{\text{train}}$. After updating the model, we test the degree to which the target triplet is disturbed by using N_{tar} as the test set to obtain the MRR value $MRR^{\text{att}}_{\text{tar}}$. Previous research evaluated the effectiveness of an attack by comparing the $MRR^{\text{att}}_{\text{tar}}$ value with the MRR value of the model before the attack. However, our evaluation model considers the influence of the added data itself and the model's generalization by only obtaining the MRR value of the target triplet N_{tar} as the test set performance without comparison. Furthermore, we use N_{con} and N_{sub} as test sets to obtain $MRR^{\text{att}}_{\text{con}}$ and $MRR^{\text{att}}_{\text{sub}}$, respectively.

Control Experiment. In the control experiment, we add the factual triplet set N_{tar} to the original training set T^{ori}_{train} to create a new training set T^{con}_{train}. The number of added samples is the same as the number of poisoned samples in the attack experiment. Note that means multiple different data poisoning experiments cannot share the same control experiment because different poisoning attacks can generate different amounts

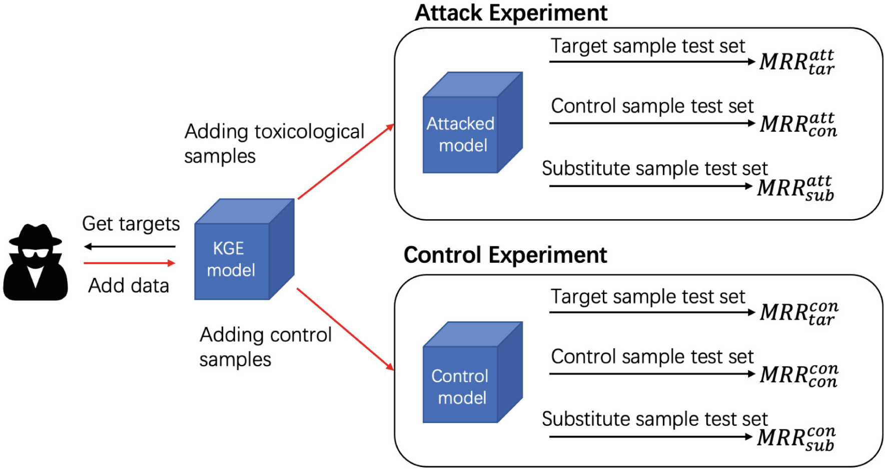


Fig. 3. The above figure shows the grouping of the attack and control experiments, as well as the indicators we need to obtain in the experiments.

of poisoned data even when using the same poisoning seed. After updating the model, we useN_{tar}, N_{con}, and N_{sub} as test sets to obtain theMRRvalues MRR_{tar}^{con}, MRR_{con}^{con}, and MRR_{sub}^{con}, respectively. The overall experimental model can be referred to the Fig. 3

3.3 Calculation of Evaluation Metrics

In order to accurately evaluate the toxicity and stealthiness of data poisoning attacks, it is necessary to take into account the impact of model generalization. This is because models with poor generalization are more likely to exhibit unstable performance when faced with new data, making it difficult to determine whether the decrease in model performance is due to the model's generalization ability or the effectiveness of the poisoning attack. To address this issue, we designed a generalization parameter λ that is suitable for this model, following the consideration of generalization ability in prior research. λ is calculated by directly using the performance differences exhibited by the model under different new data.

$$\lambda = e^{|\varDelta_g|}$$

$$\varDelta_g = MRR_{con}^{att} - MRR_{sub}^{att}$$

The value of λ is a positive correlation function of the performance difference between the control sample test set and the substitute sample test set in the attack experiment. As this difference increases, λ also increases. Stronger generalization ability results in more stable performance when facing new data, meaning that the performance decline of the target sample caused by data poisoning attacks is not closely related to the generalization ability. Conversely, weaker generalization ability results in more unstable performance when facing new data, making the target sample more likely to exhibit performance decline due to the poor generalization ability of the model itself.

Non-target samples may also be affected by the attack. The more significant the performance decline of the target sample caused by data poisoning attacks, the worse the stealthiness of the non-target samples that are interfered by the data poisoning attack. To explain this relationship, we designed the toxicity D_t and stealthiness D_s calculation methods.

$$D_t = \lambda e^{|\Delta_t|}$$

$$D_s = \frac{e^{|\Delta_s|}}{\lambda}$$

$$\Delta_t = \frac{MRR_{tar}^{con} - MRR_{tar}^{att}}{MRR_{tar}^{con}}$$

$$\Delta_s = \frac{MRR_{con}^{att} - MRR_{con}^{con}}{MRR_{con}^{con}}$$

To more accurately evaluate the effectiveness of data poisoning attacks, it is important to consider both their toxicity and stealthiness. The generalization parameter λ is used to adjust the value of D_t and D_s. In our calculation, larger values of D_t indicate stronger toxicity, while larger values of D_s indicate stronger stealthiness.

To better integrate these two measures, we propose a final evaluation score, denoted by $D - score$, which is the harmonic mean of toxicity and stealthiness.

$$D - score = \frac{1}{\frac{1}{D_t} + \frac{1}{D_s}} = \frac{2e^{|\Delta_t|+|\Delta_s|}}{e^{|\Delta_t|+|\Delta_s|} + e^{|\Delta_t|+|\Delta_s|}}$$

The harmonic mean design ensures equal consideration of toxicity and stealthiness and better represents the overall impact of data poisoning attacks. We further introduce a bias parameter α to adjust the weight of each metric in D-score. By using α to adjust the relative weight of D_t and D_s, we can adapt D-score to different attack defense scenarios. Specifically, D-score can be expressed as D_α-score, which follows the structure of the $F1$-score metric used to measure the accuracy of binary classification models. The expression of D_α-score is given as the equation above, where the weight of the toxicity metricD_t is α times that of the stealthiness metric D_s.

4 Experiments

4.1 Experimental Setups

Datasets and Models. We conducted extensive evaluation experiments on two datasets, four models, and twelve data poisoning attack methods to evaluate our proposed approach. The first dataset used in our experiments is FB15k-237, and the second is WN18RR. Table 1 displays the

We conducted comprehensive evaluation experiments on two datasets and four models, examining a total of twelve data poisoning attack methods. The twelve attack methods encompass two categories of random attacks labeled $Random - n$ (randomly adding attack samples near the target entity) and $Random - g$ (randomly adding attack samples near all entities), methods based on triple embedding gradients [14] labeled $Zhangetal.$, and three methods that leverage knowledge graph relationship pattern

Table 1. The detailed information of the WN18RR and FB15k-237 datasets, as well as the number of target triples we intend to select.

	Entities	Relations	Training	Validation	Test	Targets			
						TransE	DistMult	ComplEx	ConvE
FB15k-237	14505	237	272115	17526	20438	746	475	588	673
WN18RR	40559	11	86835	2824	2924	172	187	196	179

design [3] labeled $*** - ***$. Sym stands for attacks based on synonyms, Inv stands for attacks based on inversion relations, Com stands for attacks on synthetic relations, $turth$ stands for selecting attack samples using the lowest soft truth score, $rank$ stands for using the entity that is second only to the correct result as an attack sample, and cos stands for selecting attack samples using the maximum cosine distance.

To gain better insights into the impact of data poisoning, we specifically targeted triplets in the link prediction task where both the head and tail entities ranked within the top $Hits@10$. It is important to note that the selection of target triplets inherently influences the effectiveness of the attacks. Nonetheless, this does not hinder the comparison between different data poisoning attacks under the same target conditions. The target samples, control samples, replacement samples, and additional fact triplets used in the evaluation were sourced directly from the original test set. The number of target samples was set to be half of the number of control samples. This decision was made to account for the experimental setup where we simultaneously targeted both the head and tail entities of the target triplet. Ideally, this would result in generating twice the number of poisoned samples as there are target samples. However, in the case of semantic compositional attacks, each target sample produced four poisoned samples. To maintain consistency with other attack experiments and ensure the same ideal number of poisoned samples, we exclusively attacked the head entity of each target sample.

4.2 Experimental Results

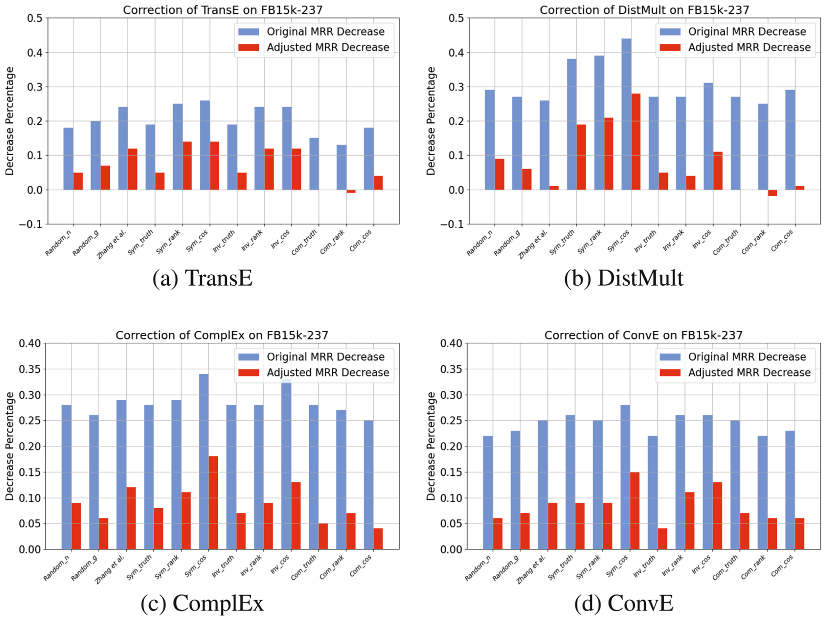
Correction of Poisoning Attack Evaluation Results. Table 2 presents the MRR values for both the attacks and control experiments. In previous studies, the original model was used as the benchmark. However, in our evaluation, we used the control model as the benchmark to calculate the percentage decrease in MRR for each model on the FB15k-237 dataset compared to the benchmark model. The results are visually depicted in Fig. 4.

Our findings indicate that on the WN18RR dataset, the control models for each attack performed similarly to the original models, suggesting that the assessment of toxicity in the past was accurate. However, on the FB15k-237 dataset, even without the addition of poisoning data, the control model exhibited a significant decrease in performance. This demonstrates that previous studies have overestimated the effectiveness of poisoning attacks.

New Evaluation Indicators. We have also introduced new evaluation indicators in Tables 3 and 4, including poisoning degree D_t, stealthiness D_s, and overall attack

Table 2. Results of Attack and Control Experiments

	TransE				DistMult				ComplEx				ConvE			
	FB15k-237		WN18RR		FB15k-237		WN18RR		FB15k-237		WN18RR		FB15k-237		WN18RR	
	Attack	Control	Attack	Control	Attack	Control	Attack	Control	Attack	Control	Attack	Control	Attack	Control	Attack	Control
$Original$	0.631	-	0.342	-	0.62	-	0.904	-	0.601	-	0.876	-	0.598	-	0.9	-
$Random - n$	0.516	0.546	0.238	0.34	0.44	0.481	0.885	0.899	0.43	0.475	0.864	0.871	0.464	0.493	0.885	0.865
$Random - g$	0.506	0.546	0.238	0.34	0.45	0.481	0.873	0.899	0.447	0.475	0.82	0.871	0.46	0.493	0.86	0.865
$Zhangetal.$	0.48	0.544	0.238	0.34	0.46	0.467	0.829	0.899	0.425	0.485	0.756	0.87	0.448	0.493	0.862	0.865
$Sym - truth$	0.514	0.54	0.532	0.338	0.386	0.478	0.664	0.906	0.431	0.471	0.581	0.87	0.445	0.489	0.586	0.875
$Sym - rank$	0.471	0.549	0.228	0.34	0.381	0.485	0.593	0.906	0.425	0.475	0.53	0.871	0.451	0.495	0.576	0.866
$Sym - cos$	0.467	0.544	0.223	0.34	0.345	0.476	0.583	0.899	0.394	0.478	0.619	0.871	0.431	0.506	0.67	0.865
$Inv - truth$	0.514	0.54	0.342	0.34	0.453	0.475	0.88	0.9	0.434	0.469	0.844	0.87	0.465	0.485	0.602	0.875
$Inv - rank$	0.482	0.549	0.239	0.34	0.453	0.47	0.86	0.899	0.43	0.475	0.815	0.871	0.444	0.497	0.581	0.866
$Inv - cos$	0.481	0.544	0.226	0.34	0.426	0.476	0.828	0.899	0.405	0.463	0.774	0.871	0.441	0.506	0.649	0.865
$Com - truth$	0.539	0.538	0.554	0.341	0.452	0.452	0.872	0.9	0.433	0.455	0.829	0.88	0.449	0.481	0.866	0.879
$Com - rank$	0.547	0.543	0.518	0.338	0.465	0.454	0.881	0.899	0.437	0.469	0.818	0.871	0.464	0.495	0.9	0.865
$Com - cos$	0.517	0.54	0.382	0.34	0.443	0.449	0.859	0.899	0.448	0.466	0.795	0.871	0.461	0.493	0.885	0.865



(a) TransE

(b) DistMult

(c) ComplEx

(d) ConvE

Fig. 4. demonstrates a significant overestimation of the effectiveness of poisoning attacks in past experiments on the FB15k-237 dataset.

effectiveness D_1, calculated based on the corrected results. We calculated the stealthiness of poisoning data quantitatively for the first time and took the influence of model generalization into consideration during the calculation process. Poisoning attacks based on semantic symmetry exhibit better poisoning degrees, and the attack samples obtained through semantic symmetry by using only correct entities in the link prediction task show excellent and stable poisoning degree on the WN18RR dataset. The poisoning degree performances on the four models were 1.400, 1.413, 1.490, and 1.435, respectively.

Table 3. Evaluated values on FB15k-237.

	TransE				DistMult				ComplEx				ConvE			
	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1
$Random - n$	1.025	1.083	0.915	0.992	1.017	1.108	0.933	1.013	1.025	1.127	0.921	1.014	1.026	1.089	0.924	1.000
$Random - g$	1.024	1.102	0.918	1.002	1.001	1.068	0.931	0.995	1.007	1.068	0.928	0.993	1.025	1.096	0.920	1.000
$Zhangetal.$	1.027	1.156	0.921	1.025	1.035	1.050	0.906	0.973	1.029	1.165	0.913	1.024	1.019	1.117	0.927	1.013
$Sym - truth$	1.028	1.079	0.934	1.001	1.030	1.249	0.922	**1.061**	1.008	1.097	**0.957**	1.022	1.026	1.123	0.905	1.003
$Sym - rank$	1.025	**1.182**	0.922	1.036	1.016	1.259	0.892	1.044	1.023	1.137	0.936	1.027	1.017	1.112	0.886	0.986
$Sym - cos$	1.016	1.171	0.933	**1.038**	1.013	**1.334**	0.863	1.048	1.024	**1.221**	0.931	**1.056**	1.022	**1.186**	0.913	1.032
$Inv - truth$	1.030	1.081	0.931	1.000	1.017	1.065	0.933	0.995	1.017	1.096	0.947	1.016	1.025	1.068	0.923	0.990
$Inv - rank$	1.027	1.161	0.915	1.023	1.018	1.056	0.917	0.981	1.011	1.112	0.945	1.022	1.022	1.137	0.932	1.025
$Inv - cos$	1.024	1.150	0.924	1.024	1.017	1.130	0.913	1.010	1.001	1.135	0.924	1.019	1.026	1.167	0.938	**1.040**
$Com - truth$	1.028	1.026	**0.941**	0.982	1.006	1.006	**0.986**	0.996	1.018	1.069	0.940	1.000	1.020	1.090	0.935	1.007
$Com - rank$	1.029	1.022	0.937	0.977	1.008	0.984	**0.988**	0.986	1.013	1.085	0.917	0.994	1.020	1.086	0.938	1.006
$Com - cos$	1.029	1.074	0.927	0.995	1.025	1.039	**1.001**	1.020	1.006	1.046	0.938	0.989	1.028	1.097	0.937	1.011

Table 4. Evaluated values on WN18RR.

	TransE				DistMult				ComplEx				ConvE			
	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1	λ	D_t	D_s	D_1
$Random - n$	1.012	1.012	0.983	0.997	1.003	1.019	0.991	1.005	1.001	1.009	1.001	1.005	1.028	1.005	0.982	0.994
$Random - g$	1.004	1.205	0.993	1.089	1.000	1.029	0.992	1.010	1.001	1.061	0.994	1.027	1.028	1.034	0.980	1.007
$Zhangetal.$	1.007	1.359	0.985	1.142	1.001	1.082	0.999	1.039	1.008	1.149	0.986	1.062	1.026	1.030	0.987	1.008
$Sym - truth$	1.007	0.567	0.985	0.720	1.006	1.314	0.991	1.130	1.006	1.402	0.986	1.158	1.025	1.427	0.980	1.162
$Sym - rank$	1.007	**1.400**	0.990	1.160	1.000	**1.413**	0.994	1.167	1.007	**1.490**	0.995	1.193	1.026	**1.435**	0.980	1.164
$Sym - cos$	1.007	1.421	0.987	1.165	1.007	1.431	0.981	1.164	1.001	1.337	0.994	1.140	1.024	1.283	0.987	1.116
$Inv - truth$	1.003	0.997	0.989	0.993	1.002	1.025	1.024	1.024	1.005	1.036	0.993	1.014	1.029	1.406	0.974	1.151
$Inv - rank$	1.008	1.357	0.984	1.141	1.005	1.050	0.993	1.020	1.002	1.069	0.991	1.028	1.024	1.423	0.985	1.164
$Inv - cos$	1.008	1.410	0.984	1.159	1.001	1.083	0.990	1.034	1.002	1.120	0.995	1.054	1.025	1.316	0.988	1.128
$Com - truth$	1.010	0.541	0.974	0.695	1.004	1.036	0.991	1.013	1.002	1.062	0.988	1.023	1.034	1.049	0.975	1.011
$Com - rank$	1.006	0.591	0.983	0.738	1.000	1.020	0.995	1.008	1.006	1.069	0.992	1.029	1.027	0.987	0.982	0.984
$Com - cos$	1.004	0.887	0.993	0.937	1.002	1.048	0.991	1.019	1.003	1.094	0.991	1.040	1.019	0.996	0.995	0.995

The stealthiness possessed by poisoning attacks based on synthetic relationships was also reflected in the value of D_s. For example, on the FB15k-237 dataset, the stealthiness of three poisoning attacks based on synthetic relationships on the DistMult model was 0.986, 0.988, and 1.001, respectively.

Generalization Variation. Figure5 illustrates the MMR difference ratio between the control group and the plain group in both the poisoning experiment and the control experiment. The convE model under attack on the WN18RR dataset is used as an example. A value of 1.0 indicates that the generalization of the poisoning model is equivalent to that of the control model. The results reveal a significant decrease in the generalization rate of the poisoning model after incorporating poisoning data into the convE model, which is observed across most poisoning attack methods. Among them, the Sym-truth and Inv-rank attacks exhibit a particularly notable impact on the model's generalization.

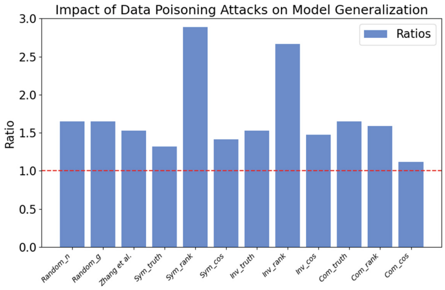


Fig. 5. Impact of Data Poisoning Attacks on Model Generalization

5 Conclusion

Evaluating the behavior of attacks is of significant research importance in studying the robustness of KGE models. However, the existing methods for evaluating attack behavior suffer from certain limitations. To address this, we propose a unified framework that enables the evaluation of various types of data poisoning attacks. Our framework encompasses key evaluation metrics such as generalizability index, toxicity, stealthiness, and combined effect. Further, we believe that the evaluation framework can be generalized to more other domains of data poisoning attacks.

References

1. Banerjee, P., Chu, L., Zhang, Y., Lakshmanan, L.V.S., Wang, L.: Stealthy targeted data poisoning attack on knowledge graphs. In: ICDE, pp. 2069–2074. IEEE (2021)
2. Bhardwaj, P., Kelleher, J.D., Costabello, L., O'Sullivan, D.: Adversarial attacks on knowledge graph embeddings via instance attribution methods. In: EMNLP (1), pp. 8225–8239. Association for Computational Linguistics (2021)
3. Bhardwaj, P., Kelleher, J.D., Costabello, L., O'Sullivan, D.: Poisoning knowledge graph embeddings via relation inference patterns. In: ACL/IJCNLP (1), pp. 1875–1888. Association for Computational Linguistics (2021)
4. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
5. Chen, R., Li, Z., Li, J., Yan, J., Wu, C.: On collective robustness of bagging against data poisoning. In: ICML. Proceedings of Machine Learning Research, vol. 162, pp. 3299–3319. PMLR (2022)
6. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI, pp. 1811–1818. AAAI Press (2018)
7. Fang, M., Sun, M., Li, Q., Gong, N.Z., Tian, J., Liu, J.: Data poisoning attacks and defenses to crowdsourcing systems. In: WWW, pp. 969–980. ACM / IW3C2 (2021)
8. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL/IJCNLP, pp. 1003–1011. The Association for Computer Linguistics (2009)
9. Steinhardt, J., Koh, P.W., Liang, P.: Certified defenses for data poisoning attacks. In: NIPS, pp. 3517–3529 (2017)

10. Tahmasebian, F., Xiong, L., Sotoodeh, M., Sunderam, V.: Crowdsourcing under data poisoning attacks: a comparative study. In: Singhal, A., Vaidya, J. (eds.) Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Regensburg, Germany, June 25–26, 2020, Proceedings, pp. 310–332. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49669-2_18
11. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML. JMLR Workshop and Conference Proceedings, vol. 48, pp. 2071–2080. JMLR.org (2016)
12. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (Poster) (2015)
13. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.: Collaborative knowledge base embedding for recommender systems. In: KDD, pp. 353–362. ACM (2016)
14. Zhang, H., et al.: Towards data poisoning attack against knowledge graph embedding. CoRR abs/1904.12052 (2019)
15. Zhang, Z., et al.: STG2P: a two-stage pipeline model for intrusion detection based on improved lightgbm and k-means. Simul. Model. Pract. Theory **120**, 102614 (2022)

Label-Guided Compressed Prototypical Network for Incremental Few-Shot Text Classification

Yongjie Wang, Minghao Hu, Xiantao Xu, Wei Luo(✉), and Zhunchen Luo

Information Research Center of Military Science, PLA Academy of Military Science,
Beijing, China
htqxjj@126.com

Abstract. Incremental few-shot text classification (IFSTC) involves the sequential classification of new classes with limited instances while preserving the discriminability of old classes, which can effectively support downstream tasks such as information extraction and knowledge graph construction. The primary objective of IFSTC is to achieve a balance between *plasticity* and *stability*, where classification models should be plastic enough to learn patterns from new classes, while also being stable enough to retain knowledge learned from previously seen classes. In previous work of incremental few-shot learning, the popular approach is to compress the base classes space to enhance model *plasticity*. However, the application of current space compression methods to text classification presents challenges due to the difficulty in simulating latent data through text data synthesis. Moreover, freezing all model parameters to maintain model stability is not a viable solution as it fails to incorporate new knowledge. To solve the problems above, we propose Label-guided Compressed Prototypical Network (LGCPN) for IFSTC, which consists of two parts. Firstly, label-guided space compression is proposed to improve model *plasticity* with text data through leveraging the information carried by labels to assist in the space compression. Secondly, few-params tuning is designed to maintain model *stability* and enable it to learn new knowledge by selectively fine-tuning few parameters. We experimented on two public datasets to evaluate the performance of our proposed method for IFSTC. The experimental results demonstrate that our method can significantly improve the accuracy of each round in the incremental stage compared to two baselines.

Keywords: few-shot · class incremental · text classification · prototypical network

1 Introduction

Incremental few-shot text classification (IFSTC) has the potential to support downstream tasks such as information extraction and knowledge graph construction by enabling models to learn from a limited amount of labeled data. IFSTC must address the challenge of *plasticity*-*stability* dilemma. Model *plasticity* refers

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 197–208, 2023.
https://doi.org/10.1007/978-3-031-44693-1_16

to its capacity to incorporate new classes, while model *stability* pertains to its ability to retain acquired knowledge. It should be noted that excessive *plasticity* may result in catastrophic forgetting of previously learned knowledge, while excessive *stability* may impede adaptation to new knowledge. Additionally, the scarcity of samples in the incremental stage may lead to overfitting, exacerbating the challenges of maintaining *stability* in the model.

Xia et al. [13] uses entailment as a solution to the challenges posed by IFSTC. However, this entails a trade-off with regards to time efficiency, and the model's stability is adversely affected. The most recent research, as presented in ForwArd Compatible Training (FACT) [16], proposes a space compression method. This approach can be leveraged to compress the known class space, thereby facilitating the classification performance of new classes during the incremental stage. However, the challenge of *plasticity*-*stability* dilemma persists. The application of FACT [16] to text classification presents two challenges.

The first challenge pertains to improving model *plasticity* in the base stage, which is complicated by the prevalence of sample-specific features in text data. This issue makes direct synthesis of sample representations prone to high levels of noise and limits the ability of synthesized samples to accurately simulate latent samples due to the influence of grammatical rules [8]. The second challenge is maintaining model *stability* while acquiring new knowledge, which is further compounded by the limited availability of training samples when encountering new classes in text classification. This difficulty makes it challenging to maintain model *stability* while also permitting the acquisition of new knowledge.

To address the challenges mentioned above, we propose Label-guided Compressed Prototypical Network (LGCPN) for IFSTC. Our method consists of two parts: label-guided space compression and few-params tuning. The former aims to improve model plasticity by utilizing pseudo-labels to assist in space compression in the base stage. The latter is designed to keep model stability and enable the model to learn new knowledge by identifying few parameters that bolster class discrimination and fine-tuning them. To evaluate the effectiveness of our proposed method for IFSTC, we conducted experiments on two widely used benchmark datasets: clinc150 and banking77. The experiment shows that our approach consistently outperformed the baseline models, achieving an average improvement of 2–3% on the clinc150 dataset and 1–2% in accuracy on the banking77 dataset. These results demonstrate the effectiveness of our proposed method in addressing the challenges of IFSTC and highlight its potential for practical applications in real-world scenarios.

2 Background

The field of incremental few-shot text classification (IFSTC) remains relatively underexplored, and there is a need to investigate approaches that can address this challenge. One potential solution is to employ space compression techniques to compress the class space without sacrificing performance. In this section, we provide an introduction to the problem of IFSTC, as well as a review of typical space compression methods that can be used to address this problem.

2.1 Incremental Few-Shot Text Classification

IFSTC is a challenging task that can be divided into two stages: the base stage and the incremental stage. In the base stage, the dataset $\mathbf{D_b} = \{(\mathbf{x_b^i}, y_b^i)\}_{i=1}^{n_b}$ comprises n_b classes, each with a sufficient number of samples for training. The training instances $\mathbf{x_i} \in \mathbb{R}^d$ in the base stage are associated with class labels $y_i \in \mathbb{Y}_b$, which provide information about the corresponding text. In cases where the labels are inadequate, descriptive alternatives may be used as substitutes. In the incremental stage, the dataset is composed of r rounds of $\{\mathbf{D_1}, \mathbf{D_2}, \ldots \mathbf{D_r}\}$, where each incremental round $\mathbf{D_k} = \{(\mathbf{x_k^i}, y_k^i)\}_{i=1}^{n_k}$ consists of n_k classes. Notably, only a limited number of samples are available for each class in the incremental stage, and datasets from previous rounds are not accessible for training.

2.2 Compressed Virtual Prototypical Network

Compressed virtual prototypical network is a two-component model that consists of a representation part and a classifier [15]. The classifier adopts cosine similarity as its metric. The class prototype is derived from the average representation of all samples that correspond to the respective class, denoted by $\Phi(\mathbf{x})$ and depicted in Formula 1. Here,$\mathbf{p_i}$ signifies the prototype of class i.

$$\boldsymbol{p_i} = \frac{1}{n} \sum_{j=1}^{|\mathbf{D^b}|} \mathbb{I}(y_j = i)\, \Phi(\mathbf{x_j}) \tag{1}$$

Denote the output of current model as $f_v(\mathbf{x}) = [\mathbf{W}, \mathbf{P}_v]^T \Phi(x)$, the objective for reserving the embedding space is:

$$L_v(\mathbf{x}, y) = L(f_v(\mathbf{x}), y) + \alpha L(\text{Mask}(f_v(\mathbf{x}), y), \hat{y}) \tag{2}$$

$$\text{Mask}(f_v(\mathbf{x}), y) = f_v(\mathbf{x}) \otimes (\mathbf{1} - \text{Onehot}(y)) \tag{3}$$

where $\hat{y} = \text{argmax}_v P_v^T \Phi(\mathbf{x}) + \mid Y_0 \mid$ is the virtual class with maximum logit. $\otimes$ is Hadamard product, $\mathbf{1}$ is an all-ones vector. The first item in Formula 2 corresponds to the vanilla training loss, which matches the output to its ground-truth label. The second term first masks out the ground-truth logit with function Mask(.,.), and then matches the rest part to the pseudo label $\hat{y}$. Since $\hat{y}$ is the virtual class with maximum logit, Formula 2 reserves the embedding space for $\hat{y}$ explicitly. Using samples from different classes such as $\{\mathbf{x_j}, y_j\}$, $\{\mathbf{x_k}, y_k\}$, the intermediate representations $g(\mathbf{x_i})$ and $g(\mathbf{x_j})$ are obtained, and the pseudo sample is linearly combined which is displayed in Formula 4.

$$\mathbf{z_k} = h\left[\beta g(\mathbf{x_i}) + (1 - \beta)\, g(\mathbf{x_j})\right] \quad i \neq j \tag{4}$$

The final representation $\mathbf{z_k}$ is obtained through the remaining hidden layers. A symmetric loss function can be constructed for the virtual instance z with the aim of reserving embedding space:

$$L_v(\mathbf{z}, \hat{y}) = L(f_v(\mathbf{z}), \hat{y}) + \alpha L(\text{Mask}(f_v(\mathbf{z}), \hat{y}), \hat{\hat{y}}) \tag{5}$$

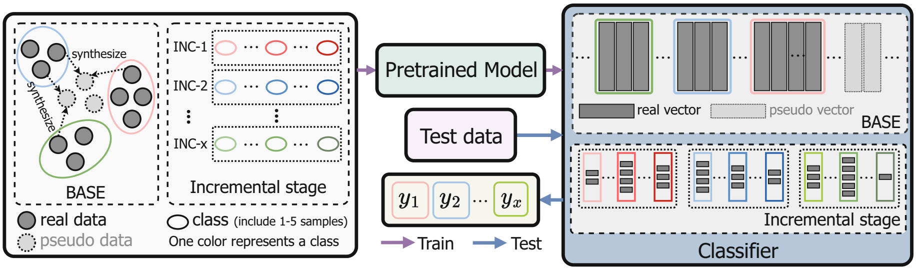


Fig. 1. An illustrative representation of the construction of a classifier in IFSTC. As depicted in the figure, class prototypes are created using both real data vectors and pseudo vectors. The representation of class prototypes in the right panel encompasses both real and virtual class prototypes, the latter of which is depicted by dotted-line rectangles. The virtual class prototypes are generated via the process of space compression illustrated in the left panel. In incremental stage, class prototypes are built from new class.

3 Label-Guided Compressed Prototypical Network

The proposed approach comprises two distinct components, namely label-guided space compression and few-params tuning. The former aims to enhance model plasticity by utilizing pseudo labels to compress space, while the latter is designed to maintain model stability and enable the model to learn new patterns through fine-tuning few parameters.

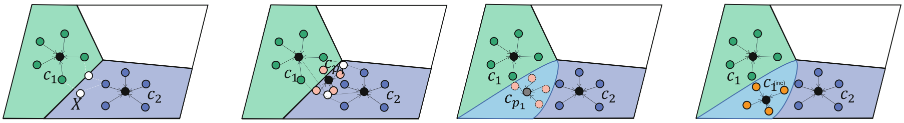
3.1 Label-Guided Space Compression

Label-guided space compression is illustrated in Fig. 2 which can be divided into three parts. The first part is label representation learning, which aims to enable the model to learn knowledge from the labels. The second part is pseudo label acquisition, which aims to acquire pseudo-labels. The third part is label guidance, which describes how pseudo labels work.

Label Representation Learning. Obtaining the representation of the label $f(\mathbf{y_i})$ during training, and using the vector value of the label representation as the vector value of the class prototype, i.e. $\mathbf{p_i} = f(\mathbf{y_i})$.

Pseudo Label Acquisition. In the base stage, we conducts a statistical analysis of the word frequencies in the label texts, selecting words with a frequency greater than 1 as pseudo labels. For instance, in the labels *Card payment wrong exchange rate* and *Compromised card*, the word *card* has a frequency greater than 1 and is thus chosen as a pseudo label. In cases where labels possess insufficient information, label description information can serve as a substitute. Apart from the virtual prototypes based on the pseudo labels, we also initializes several other virtual prototypes randomly.

Label Guidance. In the process of training the base stage, the vector representations of labels and pseudo labels are utilized as class prototypes and virtual class prototypes to construct the classifier, which is illustrate in Fig. 1. By leveraging pseudo labels, certain $\mathbf{z_k}$ in Formula 2 and Formula 5 can be attributed with greater significance.



(a) Synthesizing (b) Highlighting (c) Compressing (d) Expanding

Fig. 2. A visual illustration of the process of label-guided space compression. The process of label-guided space compression involves several steps, as depicted in the figure. Initially, pseudo samples are synthesized by using samples from distinct classes, as represented by the green and blue dots. Subsequently, favorable samples, indicated by the yellow dot, are identified by utilizing pseudo labels. The space is then compressed by employing the class center of the pseudo classes. Finally, when the model is in the incremental stage, it is expanded to incorporate additional classes. (Color figure online)

3.2 Few-Params Tuning

Few-params tuning is composed of two components, namely params selection and few epoch tuning. The former aims to select the most important parameters by utilizing the Fisher information estimation [10] while the latter aims to make the model more stable by fine-tuning few parameters over few epochs.

Params Selection. During the incremental stage, the model's capacity for classifying newly added classes can be enhanced by fine-tuning few parameters of the model. By freezing a significant portion of the pre-trained weights, the generalization ability of the model can be substantially maintained. To account for the labeled data in the incremental task, we implementaion params selection to identify the most crucial parameters for the incremental task. Specifically, we employ the Fisher information estimation [10] to identify a small subset of highly relevant parameters for a given incremental task. Formally, the Fisher Information Matrix (FIM) for the model parameters $\mathbf{W}$ is defined as follows:

$$F(\mathbf{W}) = \mathbb{E}[(\frac{\partial logp(y \mid \mathbf{x}; \mathbf{W})}{\partial \mathbf{W}})(\frac{\partial logp(y \mid \mathbf{x}; \mathbf{W})}{\partial \mathbf{W}})^T] \tag{6}$$

where $\mathbf{x}$ and y denote the input and the output respectively. The Fisher information can be interpreted as the covariance of the gradient of the log likelihood with respect to the parameters $\mathbf{W}$. EWC [5] has demonstrated the utility of diagonal elements of the Fisher information matrix (FIM) as a proxy for estimating the relevance of individual parameters to a target task. Specifically, the Fisher information, F_i, for the i-th parameter can be computed as follows:

$$F_i = \frac{1}{n} \sum_{j=1}^{n} \left(\frac{\partial \log p\left(y_i \mid \mathbf{x_j}; \mathbf{W}\right)}{\partial \mathbf{w_i}} \right)^2 \tag{7}$$

This approach is predicated on the assumption that parameters that are more salient for a given task will convey greater Fisher information. Building on this work, we propose to utilize the diagonal FIM elements to ascribe importance to individual parameters of a neural network. The parameters with higher Fisher information are posited to be more crucial for task performance. Our method thus provides a mechanism for identifying the parameters most relevant for a model's ability to execute a incremental task.

Few Epoch Tuning. Given that the computational cost associated with identifying the most important parameters is significant compared to fine-tuning during the incremental learning stage, we adopt a simplified approach whereby we identify few key parameters at the outset of the fine-tuning process and keep them fixed throughout. This strategy enables the model to adapt to the task-specific features of the new data while reducing the effects of sample noise. Additionally, we limit the fine-tuning to a small number of epochs (typically 1–2 epochs) to further minimize the risk of overfitting to the limited sample size.

4 Experiment

4.1 Dataset

We consider clinic150 [6] and banking77 [2] as our benchmark datasets to compare with previous methods, with clinic150 containing 150 classes across five domains and banking77 containing 70 classes in one domain. The clinc150 dataset was partitioned into a base of 50 classes, with a gradual inclusion of 20 classes in each round. The training samples within each class range from 1–5 samples. Additionally, a comparison was performed using 5 samples per class. And the corresponding test set comprising 30 samples per class. Similarly, the banking77 was divided into a base of 20 classes, with a gradual inclusion of 10 classes in each subsequent round.

4.2 Implementation Details

Performance Metric. In order to assess the performance of the system, the accuracy metric was employed as the performance measure. Following each training round, the average prediction accuracy for all the known classes was calculated as the key performance indicator.

Training Details. Using PyTorch as the training framework, utilizing roberta-large [1] as the pre-training model, setting the parameters to be the same as the Xia et al. [13], employing the same parameters used for compression as FACT [16], and selecting 3–6 as the parameters that can be tuned in the incremental stage.

Compared Methods. Prototypical network [9] involves using the average of the representations of the samples of each class as corresponding class prototype, which is used for expanding of the classifier. During prediction, the representations of the test samples are compared to each prototype to obtain the cosine score, the highest among which corresponds to the predicted class of the test sample. FACT [16] involves compressing the space of different class representations using virtual prototypes during the base training phase. Some virtual prototypes are randomly initialized during training. During the incremental stage, only the representations of the samples are used as the class prototype for classification, without fine-tuning.

4.3 Experiment Results

Randomly select three seeds(1, 2 and 4) and the average performance of the experimental results is shown in the table. The results presented in the table(PD denotes performance drop) indicate that the performance of FACT and prototypical network method is comparable, and the addition of label training does not result in a reduction in model classification performance. From the table, it can also be seen that directly using the compression method of FACT does not improve the classification performance of the incremental stage. After adding pseudo labels, the performance can be improved by 1 to 2 points in each round Tables 1 and 2.

The LGCPN method has proven that adding pseudo labels during the training process does not affect the performance of the model. By utilizing the obtained labels that contain general information, the base class space can be better compressed to accommodate new classes in the incremental stage. Additionally, using the method of fine-tuning with a small number of parameters can also increase the classification ability of the model to a certain extent.

4.4 Results Compared with Entailment-Based Method

The Entailment-based method is a commonly used approach for few-shot text classification that involves converting classification problems into entailment

Table 1. Accuracy of each incremental round on clinc150 dataset(%).

1-5shot		base	r1	r2	r3	r4	r5	**PD**
	Proto [9]	97.961	91.793	88.622	86.322	83.692	81.945	-16.016
	FACT [16]	97.961	92.061	88.835	86.655	83.894	82.122	−15.839
	LGCPN	**97.983**	91.840	**89.292**	**87.802**	**85.135**	**83.895**	**−14.088**
5shot								
	Proto [9]	**97.961**	92.898	90.155	88.627	86.290	84.087	−13.874
	FACT [16]	**97.961**	92.598	89.904	88.607	86.134	83.836	−14.125
	LGCPN	97.828	**93.782**	**91.609**	**90.811**	**88.166**	**86.628**	**−11.200**

Table 2. Accuracy of each incremental round on banking77 dataset(%).

1–5shot		base	r1	r2	r3	r4	r5	**PD**
	Proto [9]	**97.417**	**89.500**	83.167	**78.367**	72.514	68.274	-29.143
	FACT [16]	97.167	88.972	82.812	78.017	72.556	68.416	−28.750
	LGCPN	96.750	88.278	**83.417**	78.000	**73.222**	**69.488**	**−27.262**
5shot								
	Proto [9]	**97.417**	90.167	84.167	79.183	75.111	72.238	−25.179
	FACT [16]	97.167	90.389	84.104	78.767	74.639	71.786	−25.381
	LGCPN	96.875	**90.972**	**85.417**	**80.833**	**76.514**	**73.905**	**−22.970**

problems. Xia et al. [13] introduced this technique to the domain of incremental few-shot learning. The method proposed by Xia et al. involves pre-training the model on the MNLI dataset and constructing entailment pairs using the samples. The input text and corresponding labels of the samples are treated as positive entailment pairs, while different class labels are treated as negative entailment pairs, and the model is fine-tuned accordingly. We present a comparative analysis of LGCPN and the method proposed by Xia et al., using the

Table 3. Accuracy of each incremental round on clinc150 dataset(%).

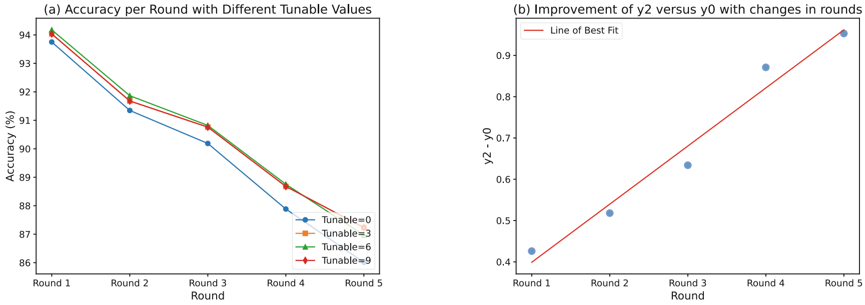
1–5shot		base	r1	r2	r3	r4	r5	**PD**
	entailment	97.600	93.095	90.593	88.576	86.385	83.440	-14.160
	LGCPN	**97.983**	91.840	89.292	87.802	85.135	**83.895**	**−14.088**
5shot								
	entailment	97.067	94.048	90.778	89.394	87.513	82.556	−14.511
	LGCPN	97.828	**93.782**	**91.609**	**90.811**	**88.166**	**86.628**	**−11.200**

clinc150 benchmark dataset. As shown in Table 3, LGCPN demonstrates superior performance on this dataset for the 5-shot learning scenario. Notably, LGCPN does not require pretraining on auxiliary datasets, enabling more rapid training. Furthermore, the training time required for Xia et al. scales linearly with the number of class labels, whereas LGCPN has a stable training time regardless of the number of classes. Taken together, these results suggest that LGCPN may be better suited for IFSTC on datasets with a large number of classes, as it can achieve strong performance without the computational overhead of pretraining and with training times that do not scale with the number of classes.

4.5 Ablation Analysis

Effectiveness of Pseudo Label. Given the complexity of the text features learned by the model, combining intermediate features synthesized by freely combining pseudo samples using FACT may lead to numerous invalid pseudo samples. In the absence of pseudo labels, the drawbacks resulting from pseudo samples may surpass their advantages, leading to an ineffective compression of the class space or even a performance decline. Pseudo labels comprise general information about the class, and their combination with the compression method presented in Formula 2 and Formula 5 may enable the learning of more meaningful general features, facilitating the improved expansion of new classes.

Few Params Tuning. Although pre-trained models have a large number of parameters, the intrinsic dimension for each downstream task is not large. In other words, theoretically we can fine-tune a very small number of parameters and still achieve good results on the incremental task. Upon varying the chosen number of parameters, the resulting impact on performance is displayed in the accompanying figure. It is discernible that the efficacy of the approach is prominent when the number of tunable parameters is set to three, and the degree of improvement is observed to increase with each subsequent round. Furthermore, the figure on the right illustrates that this performance trend is linearly additive, thereby indicating *stability* of the approach at the aforementioned parameter setting. The variable $y2$ denotes the scenario in which the number of tunable parameters is set to three, while the variable $y0$ corresponds to the case in which no tunable parameters are set, that is, the value of zero for the tunable parameters.



5 Related Work

5.1 Few-Shot Text Classification

The research on few-shot text classification primarily centers on three key areas: data augmentation, cross-task learning of general knowledge, and the extraction of prior knowledge from pre-trained models. FlipDA [17]uses a generative model and a classifier to generate label-flipped data to augment the training set. Prototypical network [9] is a meta-learning technique [3,4,11] that primarily maps a vast amount of training data to a class space, subsequently computing the prototype of each class within this space to facilitate rapid adaptation to new tasks. In prototypical network, each class corresponds to a prototype vector, which may be the average representation of all samples within that class.

5.2 Few-Shot Class Incremental Image Classification

Research on few-shot class incremental Classification is currently limited. SPPR [18] aims to enhance the scalability of the classification model by emulating the incremental stage, wherein some classes are randomly chosen from the base dataset during each round of iteration in the base training process. The margin-based approach [14] imposes a margin during base training to confine the acquisition of class-specific features, thereby promoting the assimilation of general features instead of base-class-specific ones. CEC [15] disentangles the model into two distinct components, namely, a representation module and a classifier. To facilitate class-incremental learning, graph attention networks are employed to acquire contextual information, while the generation of pseudo samples is achieved by image rotation. In FACT [16], the generation of pseudo samples is achieved through the mixing of intermediate representations of the samples, which enables the compression of the existing class space and enhances the scalability of the model, thereby facilitating the incorporation of additional incremental classes.

5.3 Incremental Few-Shot Text Classification

Xia et al. [13] proposed ENTAILMENT, which transforms text classification problems into entailment problems and pre-trains them using known entailment datasets, such as MNLI [12], and then fine-tunes them in earch round of IFSTC. Li et al. [7] introduced the adapter method in the incremental stage, which uses a small number of parameters for fine-tuning to alleviate overfitting problems with few samples, keeps the original model unchanged, and retains two samples for each class to alleviate the catastrophic forgetting. However, it still assumes 5/10 samples per class and requires 10 samples per class as a validation set for adapter.

6 Conclusion

To solve the problem of incremental text classification with few samples, we propose Label-guided Compressed Prototypical Network which effectively navigates the *plasticity-stability* dilemma. To enhance model *plasticity*, we propose label-guided space compression that utilizes pseudo labels acquired from informative label text to assist with compressing spaces. To ensure model *stability*, we introduce few-params tuning, changing the fine-tuning method of some parameters to a more suitable method for few-sample domains, and demonstrate the effectiveness of the method through experiments. For future research, there remain open questions related to our findings, including: (1) the exploration of sample selection strategies to select representative samples instead of labels, addressing the issue of inadequate content interpretability in general text classification processes; and (2) the optimization of the label synthesis strategy, such as utilizing external knowledge bases or LLM.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 62006243).

References

1. an, Y.L.: Roberta: A robustly optimized BERT pretraining approach. ArXiv preprint abs/1907.11692 (2019). https://arxiv.org/abs/1907.11692
2. Casanueva, I., Temčinas, T., Gerz, D., Henderson, M., Vulić, I.: Efficient intent detection with dual sentence encoders. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 38–45. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.nlp4convai-1.5
3. Chen, L., Jose, S.T., Nikoloska, I., Park, S., Chen, T., Simeone, O., et al.: Learning with limited samples: Meta-learning and applications to communication systems. Found. Trends® Signal Process. **17**(2), 79–208 (2023)
4. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5149–5169 (2021)
5. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
6. Larson, S., et al.: An evaluation dataset for intent classification and out-of-scope prediction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1311–1316. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1131, https://aclanthology.org/D19-1131
7. Li, G., Zhai, Y., Chen, Q., Gao, X., Zhang, J., Zhang, Y.: Continual few-shot intent detection. In: Proceedings of the 29th International Conference on Computational Linguistics., pp. 333–343 (2022)
8. Pollard, S., Biermann, A.W.: A measure of semantic complexity for natural language systems. In: NAACL-ANLP 2000 Workshop: Syntactic and Semantic Complexity in Natural Language Processing Systems (2000). https://aclanthology.org/W00-0107

9. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017(December), pp. 4–9, 2017. Long Beach, CA, USA. pp. 4077–4087 (2017), https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html
10. Tu, M., Berisha, V., Woolf, M., Seo, J., Cao, Y.: Ranking the parameters of deep neural networks using the fisher information. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20–25, 2016, pp. 2647–2651. IEEE (2016). https://doi.org/10.1109/ICASSP.2016.7472157
11. Wang, J.X.: Meta-learning in natural and artificial intelligence. Curr. Opin. Behav. Sci. **38**, 90–95 (2021)
12. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-1101, https://aclanthology.org/N18-1101
13. Xia, C., Yin, W., Feng, Y., Yu, P.: Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1351–1360. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-main.106, https://aclanthology.org/2021.naacl-main.106
14. Zou, Y.: Shanghang Zhang. Margin-based few-shot class-incremental learning with class-level overfitting mitigation, Y.L.R.L. (2022)
15. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., Xu, Y.: Few-shot incremental learning with continually evolved classifiers (2021)
16. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9046–9056 (2022)
17. Zhou, J., Zheng, Y., Tang, J., Jian, L., Yang, Z.: FlipDA: Effective and robust data augmentation for few-shot learning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8646–8665. Association for Computational Linguistics, Dublin, Ireland (2022). https://doi.org/10.18653/v1/2022.acl-long.592, https://aclanthology.org/2022.acl-long.592
18. Zhu, K., Cao, Y., Zhai, W., Cheng, J., Zha, Z.J.: Self-promoted prototype refinement for few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6801–6810 (2021)

Auxiliary Information Enhanced Span-Based Model for Nested Named Entity Recognition

Yiming Sun(✉), Chenyang Li, and Weihao Kong

School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
{sunyiming,2021100903,2022200133}@mails.cust.edu.cn

Abstract. Span-based methods have unique advantages for solving nested named entity recognition (NER) problems. As primary information, boundaries play a crucial role in span representation. However, auxiliary information, which assists in identifying entities, still needs to be adequately investigated. In this work, We propose a simple yet effective method to enhance classification performance using boundaries and auxiliary information. Our model mainly consists of an adaptive convolution layer, an information-aware layer, and an information-agnostic layer. Adaptive convolution layers dynamically acquire words at different distances to enhance position-aware head and tail representations of spans. Information-aware and information-agnostic layers selectively incorporate boundaries and auxiliary information into the span representation and maintain boundary-oriented. Experiments show that our method outperforms the previous span-based methods and achieves state-of-the-art F_1 scores on four NER datasets named ACE2005, ACE2004, Weibo and Resume. Experiments also show comparable results on GENIA and CoNLL2003.

Keywords: Named entity recognition · span-based methods · auxiliary information enhanced

1 Introduction

Named entity recognition (NER) has been regarded as a fundamental task in natural language processing. Previously, flat NER was treated as a sequence labeling that requires assigning a label to each word in a sentence accordingly [12,25,34]. This requires an assumption that the entities should be short and that there should be no overlap between them. However, in real applications, as illustrated in Fig. 1(a), an organizational noun may be nested in a personal noun. The emergence of nested entities makes the assumption no longer applicable. Therefore, it is necessary to design a model that can identify flat and nested entities. Recent methods of nested NER can be divided into four categories: 1) *sequence labeling methods* has been improved for identifying nested entities. Some

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 209–221, 2023.
https://doi.org/10.1007/978-3-031-44693-1_17

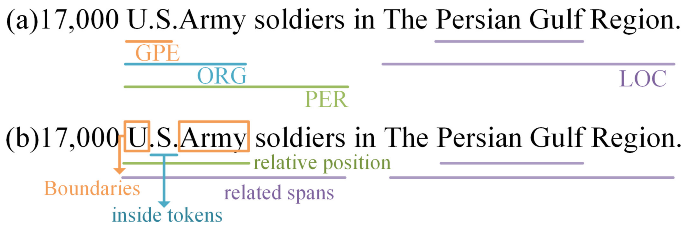

(a)17,000 U.S.Army soldiers in The Persian Gulf Region.
GPE
ORG
PER
LOC
(b)17,000 U.S.Army soldiers in The Persian Gulf Region.
relative position
Boundaries
related spans
inside tokens

Fig. 1. (a) An example sentence with nested entities from ACE2005 (b) Information that can help determine the entity.

works overlay flat NER layers [9,23] to identify nested entities. However, such practice is prone to error propagation. 2) *hypergraph-based methods* represent all entity segments as graph nodes and combine them to represent hypergraph [17]. However, such methods suffer from structural errors and structural ambiguities during inference. 3) *sequence-to-sequence methods* generate entities directly [29], which leads to inefficiencies in the decoding process and common drawbacks of sequence-to-sequence(Seq2Seq) models, such as exposure bias. 4) *span-based methods* enumerate all spans in a sentence and classify them accordingly. The approach takes the boundaries as the key to constitute the span representation [19,35]. However, only the boundaries cannot effectively detect complex nested entities [32], so focusing only on the boundaries is not comprehensive.

As shown in Fig. 1, the information available for a span to be identified includes not only the boundaries but also the auxiliary information such as inside tokens, labels, related spans, and relative positions. The utilization of the above information is helpful to solve the entity recognition problem. Although there have been works to utilize them [6,27], some issues still need to be addressed. Firstly, enumerating all possible spans in a sentence using related spans is computationally expensive. Secondly, they can only leverage part of the aforementioned auxiliary information, and most overlook relative positions' importance. Lastly, the use of related spans involves the challenge of subjective selection, which can lead to error.

In order to solve the problems mentioned above, we propose a simple but effective method to simultaneously utilize all the above-mentioned auxiliary information. The key of our model is to propose an **A**uxiliary **I**nformation **E**nhanced **S**pan-based NER (**AIESNER**) neural method. Specifically, our research follows two steps: entity extraction and entity classification. In the entity extraction stage, we design an adaptive convolution layer that contains a position-aware module, a dilated gated convolutions (DGConv) module, and a gate module. These three modules can not only dynamically acquire position-aware head and tail representations of spans by applying two single-layer fully connection layer, but also capture relationship between close and distant words. Through the acquisition of connections at different distances between words, the information-aware layer obtains auxiliary information, while the head and tail representations are used to acquire boundaries and then incorporate relatively necessary parts into span representations. Because span representations have different association strengths under different labels in the entity classification stage, we design the information-agnostic layer to apply the multi-head self-attention mechanism to establish the corresponding span-level correlation for each label. To avoid excessive attention to auxiliary information, we emphasize the boundaries at this layer with the use of only head and tail representations.

To prove the effectiveness of proposed model, we conducted experiments on six NER datasets, three of them are nested datasets, and the other three are flat datasets. For the nested datasets, proposed model achieves F_1 scores of 87.73, 87.23, and 81.40 on ACE2004, ACE2005 and GENIA, respectively. For the flat datasets, our model achieves F_1 scores of 97.07, 73.81, and 93.07 on Resume, Weibo and CoNLL2003, respectively. Using BERT as an encoder, proposed model outperforms the state-of-the-art methods on ACE2005, ACE2004, Resume and Weibo. And we get comparable results on the GENIA and CoNLL03. Our contributions are summarized as:

- This is the first work of using boundary and complete auxiliary information (i.e., inside tokens, labels, related spans, relative position) that is more efficient and reduces subjective interference.
- This work has no artificially set rules. The research does not require any external knowledge resources to achieve promising results. Thus it can be easily adapted to most usage scenarios for domain-specific data.
- The experiments explore that our proposed method performs better than the existing span-based methods and achieves state-of-the-art performance on ACE2005, ACE2004, Resume, and Weibo.

2 Related Work

2.1 Nested NER

Here we mainly focus on four nested NER methods: sequence-tagging methods, hypergraph-based methods, sequence-to-sequence methods, and span-based methods since they are similar to our work.

By stacking flat NER layers, sequence labeling methods can obtain nested entities. However, this leads to the error propagation problem. By Using dynamic stacking of flat decoding layers, [9] construct revised representations of the entities identified in the lower layers. Then provide identified entities to the next layer. Some people have improved this method by designing a reverse pyramid structure to achieve the reverse flow of information [23]. Others divide NER into two steps: merging entities and sequence labeling [4].

Hypergraph-based method was first proposed by [14] as a solution to the problem of nested entities. It has been further consolidated and enhanced by subsequent work [16,22]. The methods requires complex structures to deal with nested entities. The method also leads to structural errors and structural ambiguities during inference.

Span-based methods enumerate all span representations in a sentence and predict their types. The span representation of an entity can be obtained in various ways [11,19,31]. Several works have proposed the use of external knowledge resources. Such as the introduction of machine reading comprehension (MRC) [13] and dependency relations [10] for span prediction. The span-based methods can identify entities and their corresponding types directly [11], or they can split the process of identifying entities into two stages, including entity extraction and entity classification [19,26,27]. Compared with these previous methods, our method uses the auxiliary information that the span representation possesses.

Seq2Seq methods generate various entities directly. [5] first proposed a Seq2Seq model, where the input is the original sentence and the output is the entity start position, entity length, and entity type. [29] combines the Seq2Seq model with a BART-based pointer network. There are other methods using contrast learning [33], generative adversarial networks [8] and reinforcement learning [24] for entity recognition.

3 Approach

Figure 2 shows an overview of our approach, which consists of four main layers: the encoder layer, the adaptive convolution layer, the information-aware layer, and the information-agnostic layer.

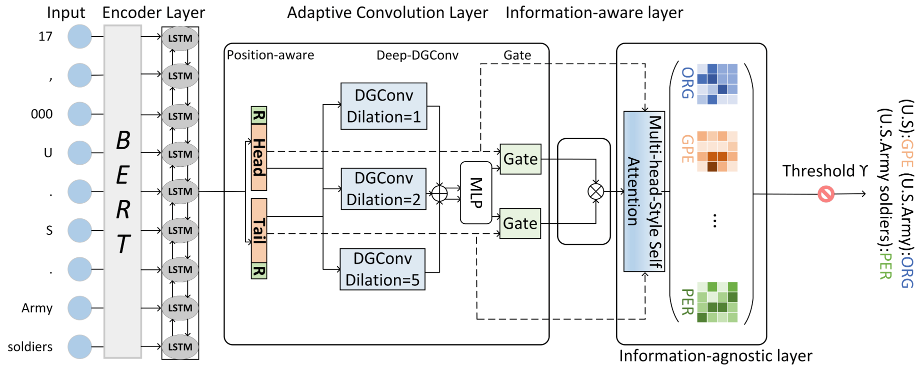


Fig. 2. The architecture of our method. MLP represents multi-layer perceptron. $\oplus$ and $\otimes$ represent concatenation and dot-product operations.

3.1 Encoder Layer

We follow [11] to encode the text. Given the input sentence $X = \{x_1, x_2, \ldots, x_N\}$ of N tokens, we first generate contextual embeddings of word pieces using BERT [3] and then combine them employing max-pooling to produce word representations. Then we adopt BiLSTM [7] to enhance the word representation. Finally, our sentence X can be represented as word representations H:

$$H = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N \times d_w} \tag{1}$$

where d_w denotes the dimension of the word representation, and N is the length of the input sentence.

3.2 Adaptive Convolution Layer

Position-Aware Module. To represent the head and tail of a span, we use two single full connection layers to transform each h_i to the head and tail vector space. At this point, we obtain the head and tail representation. In addition, position plays an essential role in identifying entities [28], so we attach position embedding from [20] to the word representation:

$$h_i^s = (W_s h_i + b_1) \otimes R_i \tag{2}$$

$$h_i^t = (W_t h_i + b_2) \otimes R_i \tag{3}$$

$$H_\delta = \left\{h_1^\delta, h_2^\delta, \ldots, h_N^\delta\right\} \tag{4}$$

where $W_s, W_t \in \mathbb{R}^{d_w \times d_h}$ and $b_1, b_2 \in \mathbb{R}^{d_h}$ are trainable parameters and bias terms, respectively. R_i is the position embedding of the i-th word, $\otimes$ is the element-wise multiplication. $\delta \in \{s, t\}$. s and t are the head and tail, respectively.

DGConv Module. We feed the head and tail representation into the same convolution module, which allows the head and tail to learn each other's word representation without introducing additional parameters. For capturing the interactions between words at different distances, we use multiple DGConv with different dilation rates r(e.g.,$r \in [1, 2, 5]$). Gated convolution avoids gradient vanishing and controls information flow while these interactions form auxiliary information. The calculation of each dilated gated convolution can be expressed as:

$$\text{DGConv}(H_\delta) = D_1 \otimes H_\delta + (\mathbf{1} - D_1) \otimes \phi\,(D_2) \tag{5}$$

$$C_\delta^r = \sigma\,(\text{DGConv}\,(H_\delta)) \tag{6}$$

where D_1 and D_2 are parameter-independent 1-dimensional convolution with H_δ as input. σ and ϕ are relu and sigmoid activation functions, respectively. $\otimes$ is element-wise multiplication, and $\mathbf{1}$ is a 1-vector with its dimension matching D_1. After that, we combine the different dilatation rates of C_δ^r to get the final result $C_\delta = \left[C_\delta^1; C_\delta^2; C_\delta^5\right] \in \mathbb{R}^{N \times 3d_h}$ and feed it into the multi-layer perceptron (MLP) to reduce the dimension:

$$Q_\delta = MLP\,(C_\delta) \in \mathbb{R}^{N \times d_h} \tag{7}$$

Gate Module. Since the previous work [6,32] demonstrated that the boundaries are practical, we balance the word representation itself with the extracted word representation at different distances. Then we can filter the unnecessary information. The gate module is shown below:

$$r_\delta = W_1 H_\delta + W_2 Q_\delta + b \tag{8}$$

$$O_\delta = r_\delta \otimes H_\delta + (\mathbf{1} - r_\delta) \otimes Q_\delta \tag{9}$$

where H_δ and Q_δ are from Eqs. 4 and 7. $W_1, W_2 \in \mathbb{R}^{d_h \times d_h}$ and $b \in \mathbb{R}^{d_h}$ are trainable parameters and bias term, respectively. $\mathbf{1}$ is a 1-vector with its dimension matching H_δ. $\otimes$ is element-wise multiplication. Finally, we get head and tail representation:

$$S = Q_s = \{s_1, \ldots, s_N\} \in \mathbb{R}^{N \times d_h} \tag{10}$$

$$T = Q_t = \{t_1, \ldots, t_N\} \in \mathbb{R}^{N \times d_h} \tag{11}$$

3.3 Information-Aware Layer

To integrate boundaries and auxiliary information into the span representation. We obtain Span(i, j) by dot product s_i^T and t_j, T is for transposition:

$$\text{Span}(i, j) = s_i^T t_j \tag{12}$$

Span$(i, j) \in \mathbb{R}^{1 \times 1}$ indicates the region of a candidate span from the i-th word to the j-th word in a sentence. Due to the filtering of the gate module [1] and the local attention of the convolution [2], the model can learn to discriminate the importance of words acquired at different distances. Thus, s_i and t_j itself will yield the boundary, close and distant words that are strongly associated with the current word pair (s_i, t_j) will be the inside tokens and related spans, respectively:

$$(A + B)^T (C + D) = A^T C + A^T D + B^T C + B^T D \tag{13}$$

Here we simplify the process and ignore the weights. As in Fig. 1, suppose the current entity Span(i, j) is [*U.S.Army*], A represents *U*, B represents *Persian*, C represents *Army*, and D represents *Gulf*. $A + B$ represents the word representation of *U* that obtains *Persian* information from upper layer. $A^T D$ represents the boundary, and $B^T D$ represents the required related spans. Thus, instead of enumerating all spans, Span(i, j) can obtain boundaries, inside tokens, and related spans, while the model can learn weights to adjust their importance. Additionally, the relative positions can be determined by using position embedding attached to the word representation. We take the boundary as an example:

$$\left(R_i h_i^s\right)^T \left(R_j h_j^t\right) = h_i^{sT} R_{j-i} h_j^t \tag{14}$$

where R_i and R_j are the position embeddings of the i-th and j-th words mentioned earlier (Eq. 2), related spans and inside tokens can also acquire their relative position.

3.4 Information-Agnostic Layer

Excess auxiliary information cluttering the span representation during entity classification may cause incorrect boundary predictions. So the boundaries become more significant in this layer. And in order to learn the correlation intensities of span representation to different labels, motivated by the multi-head self-attention mechanism, we set the number of heads as the size of entity types, then apply attention operations. We denote $\mathrm{c}_\alpha(i,j)$ as the correlation intensities of $\mathrm{Span}(i,j)$ under the α tag and only use the boundaries to form span representation.

$$\mathrm{c}_\alpha(i,j) = W_\alpha^T \left[h_i^s; h_j^t\right] \tag{15}$$

where $\alpha \in \{1,2,\ldots,|T|\}$, $|T|$ is the number of labels. $W_\alpha \in R^{(2\times d_h)}$ is the trainable parameters. [;] means concatenation operation. h_i^s and h_j^t are from Eq. 2 and Eq. 3. We combine the results of entity extraction and entity classification to get the final span score:

$$p_{i,j}^\alpha = \mathrm{Span}(i,j) + \mathrm{c}_\alpha(i,j) \tag{16}$$

3.5 Training and Inference Details

During training, we follow [21] which generalizes the softmax cross-entropy loss to multi-label classification. The method effectively solved the problem of positive and negative label imbalance. In addition, as in [31], we set a threshold γ to determine whether span belongs to label α. The loss function can be formulated as follows:

$$\mathcal{L}_\alpha = \log\left(e^\gamma + \sum_{(i,j)\in\Omega_\alpha} e^{-p_{i,j}^\alpha}\right) + \log\left(e^\gamma + \sum_{(i,j)\notin\Omega_\alpha} e^{p_{i,j}^\alpha}\right) \tag{17}$$

where Ω_α represents the set of entities span belonging to label α, γ is set to 0. Finally, we add up the loss on all labels to get the total loss:

$$\mathcal{L} = \sum_{\alpha\in\varepsilon} \mathcal{L}_\alpha \tag{18}$$

where $\varepsilon = \{1, 2, ..., |T|\}$, $[T]$ is the number of labels.

During inference, The span satisfying $p_{i,j}^\alpha > 0$ is the output of the entity belonging to the label α.

4 Experiments

4.1 Datasets

To evaluate the performance of our model on the two NER subtasks, we conduct experiments on six datasets.

Flat NER Datasets. We conduct experiments on the English dataset CoNLL2003 and the Chinese dataset Resume and Weibo. We employ the same experimental setting in previous work [29].

Nested NER Datasets We conducted experiments on the GENIA, ACE2005, and ACE2004. For ACE2005 and ACE2004, we used the same dataset split as [14]. For GENIA, we followed [11] using five types of entities, dividing the train/dev/test as 8.1:0.9:1.0.

Table 1. Results for flat NER datasets. † represents our re-implementation with their code.

	CoNLL2003			Weibo			Resume		
	P	R	F1	P	R	F1	P	R	F1
Span-based Methods									
Locate and Lable [19]	92.13	**93.70**	92.94	70.11	68.12	69.16	–	–	–
W2NER [11]	92.71	93.44	**93.07**	70.84	73.87	72.32	96.96	96.35	96.65
Biaffine [31] †	92.91	92.13	92.52	–	–	–	–	–	–
Baseline+BS [35]	–	–	–	70.16	**75.36**	72.66	96.63	**96.69**	96.66
Others									
TENER [28]	–	–	–	–	–	58.17	–	–	95
LSTM + Lexicon augment [15]	–	–	–	70.94	67.02	70.5	96.08	96.13	96.11
FLAT [12]	–	–	–	–	–	60.32	–	–	95.45
AESINER [18]	–	–	–	–	–	69.78	–	–	96.62
BartNER+BART [29] †	92.56	93.56	93.05	–	–	–	–	–	–
AIESNER(Ours)	**93.08**	93.06	**93.07**	**74.45**	73.19	**73.81**	**97.58**	96.56	**97.07**

Table 2. Results for nested NER datasets.

	ACE2004			ACE2005			GENIA		
	P	R	F1	P	R	F1	P	R	F1
Span-based Methods									
Briaffine [31]	87.30	86.00	86.70	85.20	85.60	85.40	81.80	79.30	80.50
A Span-based Model [10]	–	–	–	–	–	83.00	–	–	77.80
Locate and Lable [19]	87.44	87.38	87.41	86.09	87.27	86.67	80.19	80.89	80.54
Triaffine [32]	87.13	87.68	87.40	86.7	86.94	86.82	80.42	**82.06**	81.23
CNN-NER [30]	**87.82**	87.40	87.61	86.39	87.24	86.82	**83.18**	79.7	**81.40**
W2NER [11]	87.33	**87.71**	87.52	85.03	**88.62**	86.79	83.1	79.76	81.39
Others									
SH+LSTM [22]	78.00	72.40	75.10	76.80	72.30	74.50	77.00	73.30	75.10
Neural layered model [9]	–	–	–	74.20	70.30	72.20	78.50	71.30	74.70
BartNER+BART [29]	87.27	86.41	86.84	83.16	86.38	84.74	78.87	79.6	79.23
SMHSA [27]	86.90	85.80	86.30	85.70	85.20	85.40	80.30	78.90	79.60
AIESNER(ous)	**87.82**	87.64	**87.73**	**86.97**	87.49	**87.23**	81.75	81.06	**81.40**

4.2 Results for Flat NER

We evaluate our model on CoNLL03, Weibo, and Resume. As shown in Table 1, F_1 scores of our model were 93.07, 73.81 and 97.07, respectively, outperforming the representatives of other methods (+0.02 on CoNLL2003, +3.31 on Weibo, +0.45 on Resume). Compared to other span-based methods, our model achieves the best performance on the F_1 scores of Resume (+0.41 vs. baseline+BS) and Weibo (+1.14 vs. baseline+BS), reaching the state-of-the-art results and on CoNLL03 (+0.00 vs. W2NER) we also achieved competitive results. Further-

Table 3. Model ablation studies F_1. DGConv(r=1) denotes the convolution with the dilation rate 1. "-" means remove the module.

	ACE2005	Weibo	Genia
Ours	**87.23**	**73.81**	**81.40**
-Gate Module	86.74 (−0.49)	71.32 (−2.49)	80.98 **(−0.42)**
Gate replaced with Add	86.44 (**−0.79**)	70.56 (**−3.25**)	81.08 (−0.32)
DGConv replaced with DConv	86.71 (−0.52)	73.06 (−0.75)	81.05 (−0.35)
-Position Emb	86.64 (−0.59)	72.46 (−1.35)	80.33 **(−1.07)**
-DGConv	86.48 **(−0.75)**	71.38 **(−2.43)**	80.68 (−0.72)

more, our model achieves the best precision performance, demonstrating the effectiveness of the auxiliary information we incorporated.

4.3 Results for Nested NER

Table 2 shows the performance of our model on ACE2004, ACE2005, and GENIA. F_1 scores of our model were 87.73, 87.23, and 81.40, respectively, which substantially outperforms the representatives in other methods (+0.89 on ACE2004, +1.83 on ACE2005, +0.80 on GENIA), proving the advantage of span-based methods in solving nested NER. Compared with other span-based methods, our model outperforms previous state-of-the-art methods in terms of F_1 scores for ACE2004 (+0.12 vs. CNN-NER) and ACE2005 (+0.41 vs. Triaffine). Our model also achieved competitive performance for GENIA (+0.00 vs. CNN-NER).

4.4 Ablation Study

As shown in Table 3, we ablate or replace each part of the model on ACE2005, Weibo, and GENIA. First, we remove the gate module, and the performance drop proves the importance of the boundaries. In contrast, changing the gates to direct addition would make the model unable to use the information obtained selectively. The overall performance drop is more pronounced than the weakening of the boundaries information. The model's performance drops after replacing the DGconv with the Dconv. After removing the adaptive convolution layer or position embedding, the performance of the model decreases significantly.

4.5 Case Study

To analyze the effectiveness of auxiliary information, we show two examples from the ACE2005 and GENIA datasets in Table 4. We remove the position embedding, DGConv module, and Gate module from the dynamic convolution layer to eliminate the effect of auxiliary information. In the first example, the model misclassifies "*Sukhoi Su-27*" as "None" and "*Su-27*" as "VEH" in the absence

Table 4. Case study on ACE2005 and GENIA dataset. The colored brackets indicate the boundary and label of the entity. "AUX infor" is the abbreviation for auxiliary information.

		AIESNER w/o AUX infor		AIESNER	
Span	Gold label	label	$p^{\alpha}_{i,j}$	label	$p^{\alpha}_{i,j}$
. . . [several squadrons of [Sukhoi Su-27]$_{\text{VEH}}$ interceptors, considered [[the world]$_{\text{LOC}}$'s premier dogfighters]$_{\text{VEH}}$]$_{\text{VEH}}$.					
several squadrons of Sukhoi Su-27 . . . dogfighters	VEH	VEH	2.00	VEH	7.08
Sukhoi Su-27	VEH	None	−0.29	VEH	4.98
the world	LOC	LOC	5.95	LOC	8.12
the world's premier dogfighters	VEH	None	−1.56	VEH	2.41
Su-27	None	VEH	3.79	None	−3.89
. . . [octamer element]$_{\text{DNA}}$ 5-ATGCAAAG-3, located in the [upstream region]$_{\text{DNA}}$ of this [promoter]$_{\text{DNA}}$ and in the [promoters]$_{\text{DNA}}$ of ...					
octamer element	DNA	DNA	0.94	DNA	2.79
upstream region	DNA	DNA	1.21	DNA	2.19
promoter	DNA	None	−0.17	DNA	1.92
promoters	DNA	DNA	0.29	DNA	2.49

of auxiliary information. However, with the help of auxiliary information, the model corrects them to "VEH" and "None". In the second example, the model successfully corrects "*promoter*" from the "None" to the "DNA". In addition, with the help of the auxiliary information, the confidence level $p^{\alpha}_{i,j}$ of the model for the correct label can be significantly improved.

5 Conclusion

In this paper, we propose a span-based method for nested and flat NER. We argue that boundaries and auxiliary information including relative position, inside tokens, labels, and related spans, should be used reasonably to enhance span representation and classification. To this end, we design a model that automatically learns the correlation between boundaries and auxiliary information, avoiding the error and tedium of human-defined rules. Experiments show that our method outperforms all span-based methods and achieves state-of-the-art performance on four datasets.

Acknowledgement. This work was supported by the Jilin Provincial Department of Education Science and Technology Research Planning Project, Grant number jjkh20220779kj. Jilin Provincial Science and Technology Development Plan Project, Grant number 20220201149gx.

References

1. Cao, H., et al.: OneEE: a one-stage framework for fast overlapping and nested event extraction. In: Proceedings of COLING, pp. 1953–1964. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (2022)
2. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. arXiv preprint arXiv:1911.03584 (2019)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
4. Fisher, J., Vlachos, A.: Merge and label: a novel neural network architecture for nested NER. In: Proceedings of ACL, pp. 5840–5850. Association for Computational Linguistics, Florence, Italy (2019)
5. Gillick, D., Brunk, C., Vinyals, O., Subramanya, A.: Multilingual language processing from bytes. In: Proceedings of NAACL, pp. 1296–1306. Association for Computational Linguistics, San Diego, California (2016)
6. Gu, Y., Qu, X., Wang, Z., Zheng, Y., Huai, B., Yuan, N.J.: Delving deep into regularity: A simple but effective method for Chinese named entity recognition. In: Findings of NAACL, pp. 1863–1873. Association for Computational Linguistics, Seattle, United States (2022)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Huang, P., Zhao, X., Hu, M., Fang, Y., Li, X., Xiao, W.: Extract-select: a span selection framework for nested named entity recognition with generative adversarial training. In: Findings of ACL, pp. 85–96. Association for Computational Linguistics, Dublin, Ireland (2022)
9. Ju, M., Miwa, M., Ananiadou, S.: A neural layered model for nested named entity recognition. In: Proceedings of NAACL, pp. 1446–1459. Association for Computational Linguistics, New Orleans, Louisiana (2018)
10. Li, F., Lin, Z., Zhang, M., Ji, D.: A span-based model for joint overlapped and discontinuous named entity recognition. In: Proceedings of ACL-IJCNLP, pp. 4814–4828. Association for Computational Linguistics, Online (2021)
11. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 10965–10973 (2022)
12. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: Proceedings of ACL, pp. 6836–6842. Association for Computational Linguistics, Online (2020)
13. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified MRC framework for named entity recognition. In: Proceedings of ACL, pp. 5849–5859. Association for Computational Linguistics, Online (2020)
14. Lu, W., Roth, D.: Joint mention extraction and classification with mention hypergraphs. In: Proceedings of EMNLP, pp. 857–867. Association for Computational Linguistics, Lisbon, Portugal (2015)
15. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of ACL, pp. 5951–5960. Association for Computational Linguistics, Online (2020)
16. Muis, A.O., Lu, W.: Learning to recognize discontiguous entities. In: Proceedings of EMNLP, pp. 75–84. Association for Computational Linguistics, Austin, Texas (2016)

17. Muis, A.O., Lu, W.: Labeling gaps between words: recognizing overlapping mentions with mention separators. In: Proceedings of EMNLP, pp. 2608–2618. Association for Computational Linguistics, Copenhagen, Denmark (2017)
18. Nie, Y., Tian, Y., Song, Y., Ao, X., Wan, X.: Improving named entity recognition with attentive ensemble of syntactic information. In: Findings of EMNLP, pp. 4231–4245. Association for Computational Linguistics, Online (2020)
19. Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., Lu, W.: Locate and label: a two-stage identifier for nested named entity recognition. In: Proceedings of ACL-IJCNLP, pp. 2782–2794. Association for Computational Linguistics, Online (2021)
20. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864 (2021)
21. Sun, Y., et al.: Circle loss: a unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6398–6407 (2020)
22. Wang, B., Lu, W.: Neural segmental hypergraphs for overlapping mention recognition. In: Proceedings of EMNLP, pp. 204–214. Association for Computational Linguistics, Brussels, Belgium (2018)
23. Wang, J., Shou, L., Chen, K., Chen, G.: Pyramid: a layered model for nested named entity recognition. In: Proceedings of ACL, pp. 5918–5928. Association for Computational Linguistics, Online (2020)
24. Wang, X., et al.: Automated concatenation of embeddings for structured prediction. In: Proceedings of ACL-IJCNLP, pp. 2643–2660. Association for Computational Linguistics, Online (2021)
25. Wang, X., et al.: Improving named entity recognition by external context retrieving and cooperative learning. In: Proceedings of ACL, pp. 1800–1812. Association for Computational Linguistics, Online (2021)
26. Wang, Y., Yu, B., Zhu, H., Liu, T., Yu, N., Sun, L.: Discontinuous named entity recognition as maximal clique discovery. In: Proceedings ACL-IJCNLP, pp. 764–774. Association for Computational Linguistics, Online (2021)
27. Xu, Y., Huang, H., Feng, C., Hu, Y.: A supervised multi-head self-attention network for nested named entity recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14185–14193 (2021)
28. Yan, H., Deng, B., Li, X., Qiu, X.: Tener: adapting transformer encoder for named entity recognition. arXiv preprint arXiv:1911.04474 (2019)
29. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various NER subtasks. In: Proceedings of ACL-IJCNLP, pp. 5808–5822. Association for Computational Linguistics, Online (2021)
30. Yan, H., Sun, Y., Li, X., Qiu, X.: An embarrassingly easy but strong baseline for nested named entity recognition. arXiv preprint arXiv:2208.04534 (2022), https://arxiv.53yu.com/pdf/2208.04534
31. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of ACL, pp. 6470–6476. Association for Computational Linguistics, Online (2020)
32. Yuan, Z., Tan, C., Huang, S., Huang, F.: Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition. In: Findings of ACL, pp. 3174–3186. Association for Computational Linguistics, Dublin, Ireland (2022)
33. Zhang, S., Cheng, H., Gao, J., Poon, H.: Optimizing bi-encoder for named entity recognition via contrastive learning. arXiv preprint arXiv:2208.14565 (2022)

34. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Proceedings of ACL, pp. 1554–1564. Association for Computational Linguistics, Melbourne, Australia (2018)
35. Zhu, E., Li, J.: Boundary smoothing for named entity recognition. In: Proceedings of ACL, pp. 7096–7108. Association for Computational Linguistics, Dublin, Ireland (2022)

DialogRE^{C+}: An Extension of DialogRE to Investigate How Much Coreference Helps Relation Extraction in Dialogs

Yiyun Xiong[1], Mengwei Dai[1], Fei Li[1], Hao Fei[2], Bobo Li[1], Shengqiong Wu[2], Donghong Ji[1], and Chong Teng[1](✉)

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
2018302180072@whu.edu.cn

[2] School of Computing, National University of Singapore, Singapore, Singapore

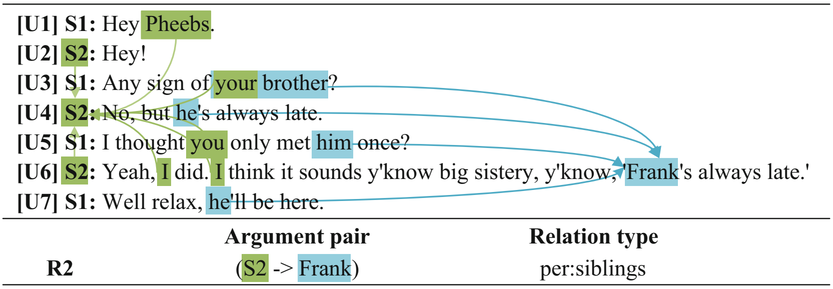
Abstract. Dialogue relation extraction (DRE) that identifies the relations between argument pairs in dialogue text, suffers much from the frequent occurrence of personal pronouns, or entity and speaker coreference. This work introduces a new benchmark dataset DialogRE^{C+}, introducing coreference resolution into the DRE scenario. With the aid of high-quality coreference knowledge, the reasoning of argument relations is expected to be enhanced. In DialogRE^{C+} dataset, we manually annotate total 5,068 coreference chains over 36,369 argument mentions based on the existing DialogRE data, where four different coreference chain types namely speaker chain, person chain, location chain and organization chain are explicitly marked. We further develop 4 coreference-enhanced graph-based DRE models, which learn effective coreference representations for improving the DRE task. We also train a coreference resolution model based on our annotations and evaluate the effect of automatically extracted coreference chains demonstrating the practicality of our dataset and its potential to other domains and tasks.

Keywords: Dialogue relation extraction · Coreference resolution · Graph neural network

1 Introduction

Relation extraction (RE) is a long-standing task in natural language processing (NLP) community, aiming to detect semantic relationships between two arguments in one sentence [5,30], or multiple sentences (i.e., document) [14,23,26]. The latest research interest of RE has been shifted from the sentence level to the dialogue level, which identifies the relations between pairs of arguments in the conversation context. As can be exemplified in Fig. 1, DRE often comes with cross-sentence relations due to the characteristic of multiple parties and threads within dialogue context [27]. This makes DRE much more challenging compared with the common sentence-level RE [25]

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 222–234, 2023.
https://doi.org/10.1007/978-3-031-44693-1_18



[U1] S1: Hey Pheebs.
[U2] S2: Hey!
[U3] S1: Any sign of your brother?
[U4] S2: No, but he's always late.
[U5] S1: I thought you only met him once?
[U6] S2: Yeah, I did. I think it sounds y'know big sistery, y'know, 'Frank's always late.'
[U7] S1: Well relax, he'll be here.

	Argument pair	Relation type
R2	(S2 -> Frank)	per:siblings

Fig. 1. An example dialogue and its corresponding relations in DialogRE [27]. S1, S2 denotes an anoymized speaker of each utterance ([U*]). The words marked in green and blue are the mentions of subjective argument 'S2' and objective argument 'Frank'.

The primary challenge in DRE that causes difficulty to understand dialogue semantics lies in the **speaker coreference problem** [29]. In the dialogue scenario, speakers often use pronouns (e.g., 'he', 'she', 'it') for referring to certain targets, instead of the target names, such as person, location and organization names. Especially in DialogRE [27], personal pronouns are frequently witnessed, which greatly hinder the relation reasoning of the task. As illustrated in Fig. 1, Speaker2 (S2) are coreferred with various pronouns in different utterances (e.g., 'you' in [U5], 'I' in [U6]) refering to one identical person 'Pheebs'; another person 'Frank' are referred as 'your brother' in [U3] and 'him' in [U5] by different speakers. Without correctly reasoning the coreference of speakers or their roles, it will be problematic to understand or infer the relations between arguments. Unfortunately, previous DRE works either ignore the coreference information [11,19] or utilize inaccurate or incomplete coreference information extracted based on heuristic rules [4,16,29].

To this end, this work contributes to DRE with a new dataset **DialogRE**$^{C+}$, where all the coreference chains [18] are annotated manually by 11 graduate students and based on the existing DialogRE data. To facilitate the utility of DialogRE^{C+}, we define four types of coreference chains, including *Speaker Chain*, *Person Chain*, *Location Chain* and *Organization Chain*. Finally, DialogRE^{C+} marks 36,369 mentions involved in total 5,068 coreference chains.

Based on the DialogRE^{C+} dataset, we develop 4 coreference-enhanced graph-based DRE models, in which the coreference features are properly modeled and represented for learning comprehensive representations of arguments and better reasoning of the argument relations. In addition, in order to explore the improvement effect of automatically extracted coreference chains on DRE, we train a coreference resolution model [12] using the English coreference resolution data from the CoNLL-2012 shared task [18] and our DialogRE^{C+}, and then employ extracted coreference information in the DRE models.

Experimental results show that the inclusion of coreference chains in our DialogRE^{C+} dataset has substantially enhanced the performance of each model, compared to their original counterparts [2,11,28,29]. Specifically, this improvement is reflected in average F1 score increases of 2.8% and 3.2% on the development and test sets, respectively. Moreover, the automatically extracted coreference chains improve 1.0% and 0.6% F1s on average compared with original models. Further analysis demonstrates that the method, when augmented with annotated coreference information, exhibits superior performance in detecting cross-utterance relations. We release the DialogRE^{C+} dataset and the benchmark models to facilitate subsequent research.[1][2]

2 Related Work

2.1 Relation Extraction

Intra-/Inter-Sentence RE. Relation extraction is one of the key tracks of information extraction [1,6,7,13]. Most of the previous RE research focus on the sentence-level relation extraction, predict the relationships between two entities within a single sentence with neural network modeling [8,30]. Due to the fact that a large number of relations are expressed in multiple sentences in practice, the extraction scope has been expanded to the inter-sentence scenario. Nan et al. [17] empowered the relational reasoning across sentences by automatically inducing the latent document-level graph, and develop a refinement strategy to incrementally aggregate relevant information. Tang et al. [22] proposed a hierarchical inference network by considering information from entity, sentence, and document levels. GAIN [28] constructs two graphs to capture complex interactions among different mentions underlying the same entities.

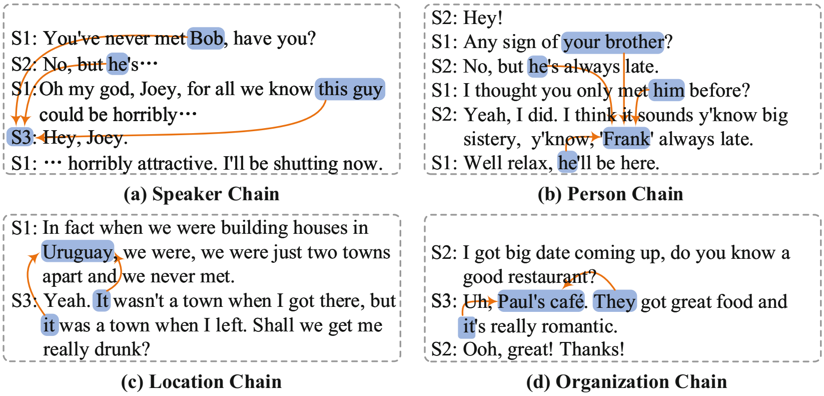
Dialogue-Level RE. In 2020, Yu et al. [27] propose dialogue-based relation extraction dataset (DialogRE) and the DRE task on this basis. REDialog [29] designs speaker embeddings and speaker-pronoun coreference particularly for the features of dialogue text, and word-relation attention and graph reasoning are used to further enhance the model. TUCORE-GCN [11] is an utterance context-aware graph convolutional network. A heterogeneous dialogue graph is introduced to model the interaction between arguments in the whole dialogue. HGAT [2] presents a graph attention network-based method where a graph that contains meaningfully connected speaker, entity, type, and utterance nodes is constructed. Fei et al. [4] construct dialogue-level mixed dependency graph (D^{2}G) for DRE with various conversational structure features, such as including dialogue answering structure, speaker coreference structure, syntactic dependency structure, and speaker-predicate structure. D^{2}G has verified the explicit integration of speaker coreference helps MRE. However, these models do not fully utilize coreference information, which limits their ability to infer argument relations.

2.2 Applications of Coreference Resolution

Coreference resolution is a core linguistic task that aims to find all expressions which refer to the same entity. Lee et al. [12] proposed the first end-to-end neural coreference resolution system. Based on this, many transformer-based models

[1] https://github.com/palm2333/DialogRE_coreference.

[2] https://gitee.com/yyxiong715/DialogRE_coreference.


S1: You've never met Bob, have you?
S2: No, but he's⋯
S1: Oh my god, Joey, for all we know this guy could be horribly⋯
S3: Hey, Joey.
S1: ⋯ horribly attractive. I'll be shutting now.
S2: Hey!
S1: Any sign of your brother?
S2: No, but he's always late.
S1: I thought you only met him before?
S2: Yeah, I did. I think it sounds y'know big sistery, y'know, 'Frank' always late.
S1: Well relax, he'll be here.
S1: In fact when we were building houses in Uruguay, we were, we were just two towns apart and we never met.
S3: Yeah. It wasn't a town when I got there, but it was a town when I left. Shall we get me really drunk?
S2: I got big date coming up, do you know a good restaurant?
S3: Uh, Paul's café. They got great food and it's really romantic.
S2: Ooh, great! Thanks!

(a) Speaker Chain **(b) Person Chain** **(c) Location Chain** **(d) Organization Chain**

Fig. 2. The illustration of four types of coreference chains.

[9,10] achieved remarkable success on the CoNLL benchmark [18]. In recent years, long text-based tasks have become increasingly abundant. Researchers have noticed the importance of coreference information and applied coreference resolution to many downstream tasks.

Wu et al. [24] developed an effective way to use naturally occurring coreference phenomena from existing coreference resolution datasets when training machine reading comprehension models. Xue et al. [25] imitated the reading process of humans by leveraging coreference information when dynamically constructing a heterogeneous graph to enhance semantic information. Coreference resolution is also applied in machine translation [15], summarization [20], dialogue [21] to improve the performance of the task. In this work, we incorporate the coreference information obtained from a coreference resolution model into DRE models to explore its impact on the DRE task.

3 DialogRE^{C+}: An Extension of DialogRE with Coreference Annotations

3.1 Annotation Method

In the annotation process, we employ the Brat annotation tool[3], which is widely applied to annotate events, entities, relationships, attributes, etc. Given a dialogue text, we annotate the pronouns that refer to the same argument or concept with coreference relations, including personal pronouns, possessive pronouns, and names. We predefine four types of coreference chains: *Speaker Chain*, *Person Chain*, *Location Chain*, and *Organization Chain*.

Speaker Chain refers to the coreference chain of people who are currently involved in communication[4]. We focus on annotating the personal pronouns and

[3] http://brat.nlplab.org.

[4] Plural personal pronouns such as "we, us, them and they" refer to multiple entities or speakers, thereby they involve relationship extraction of multiple entities. In this paper, since focus on relationship extraction of two entities, we do not mark them for the time being.

Table 1. The statistics for DialogRE^{C+}.

	Train	Dev	Test
Speaker Chain	2,277	748	784
Person Chain	645	225	232
Location Chain	48	20	37
Organization Chain	26	8	18
Mentions	21,990	7,183	7,196
Coref. Chains	2,996	1,001	1,071
Dialogues	1,073	358	357
Utterances	14,024	4,685	4,420
Argument pairs	5,997	1,914	1,862

names which refer to the same speaker when communicating, such as ‘I’, ‘you’, ‘he’, ‘she’, etc. As shown in Fig. 2.(a), we notice that the pronouns, ‘Bob’, ‘he’, ‘this guy’, refers to the same speaker, S3. Thus, the final speaker chain built is $[S3 \leftarrow (Bob, he, this\ guy)]$.

Person Chain refers to the coreference chain of people who are not currently involved in communication. It is common that people discuss another person who does not appear in the dialogue, i.e., a third entity. Person chain marks all pronouns and names referring to the same third entity. For instance, in Fig. 2.(b), we find that the two speakers, S1 and S2, are talking about ‘Frank’ who does not appear in the conversation. Thus, we mark the person chain of ‘Frank’ as $[Frank \leftarrow (your\ brother, he, him, he)]$[5].

Location Chain refers to the coreference chain of location, in which we annotate all names and pronouns of a place, such as “it”, “this”, etc. In Fig. 2.(c), S1 mentioned a place, ‘Uruguay’, and then S3 use the pronoun, ‘it’, to represent the aforementioned place. Finally, the marked location chain is $[Uruguay \leftarrow (it, it)]$.

Organization Chain represents the coreference chain of organization, in which we annotate all names and referential pronouns of an organization that is discussed in a dialogue. In Fig. 2.(d), ‘Paul’s Café’ is a new organization. In the following, ‘They’ and ‘it’ both refer to ‘Paul’s Café’. Thus, the organization coreference chain in this dialogue is $[Paul's\ Café \leftarrow (They, it)]$.

3.2 Annotation Quality Control

Before annotating, we have designed detailed annotation instructions. Then 11 graduate students are employed to annotate all the data. After data annotation is completed, a senior annotator has examined all annotated data. If there is any

[5] The two pronouns, ‘he’, appear in the third and sixth utterance, respectively.

contradiction, the senior annotator will discuss it with the corresponding annotator and reach a consensus. In this way, we ensure that at least two annotators agree on each annotation result, achieving annotation consistency.

3.3 Data Statistics

We annotate coreference chains based on the DialogRE dataset, which has a total of 36 relation types, 1,788 dialogues, 23,129 utterances, and 9,773 argument pairs, as shown in Table 1. In addition, we totally annotated 5,068 coreference chains and 36,369 mentions. In other words, each chain contains about 7 (36369/5068) mentions.

4 Backbone Models for DRE and Coreference Resolution

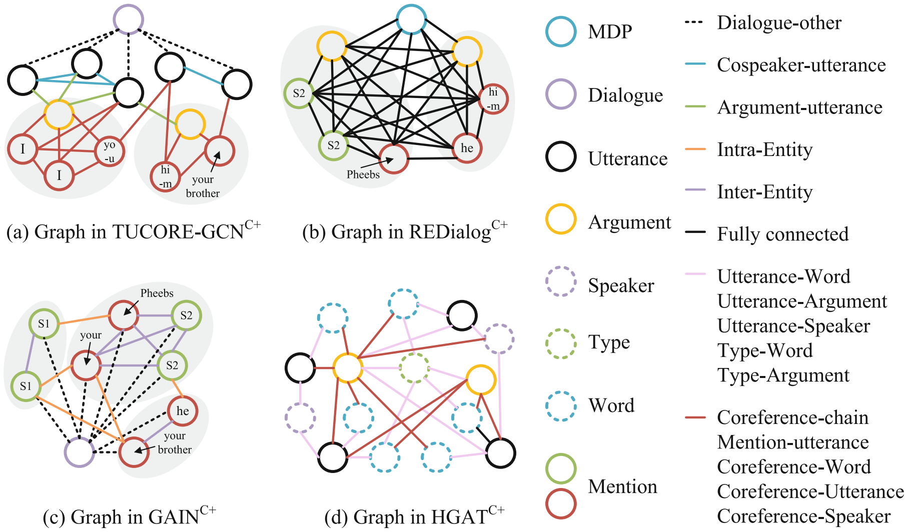
Graph-based models are widely used in the DRE task, as they can structure complicated syntactic and semantic relations. Therefore, we choose four graph-based models and enrich the dialogue graph using coreference chains to investigate how much coreference helps RE. Figure 3 shows the graph structure of these DRE models.[6] It is worth noting that the red nodes are the nodes we propose or newly added mention nodes, and the red edges are our newly proposed edges. In order to verify the effect of automatically extracted coreference information, we train a coreference resolution model, and add the machine-predicted coreference chains to the above graph-based models.

4.1 DRE Models

TUCORE-GCN^{C+} encodes the dialogue text with BERT$_s$ [27], and then applies a masked multi-head self-attention to effectively extract the contextualized representation of each utterance from BERT$_s$. Next, as shown in Fig. 3.(a), we build a heterogeneous graph over the pre-defined nodes and edges. Thereafter, GCN is adopted to model the heterogeneous graph, resulting in a surrounding utterance-aware representation for each node. Finally, we inference the relations between each argument pair based on these coreference-enhanced features.

The coreference-enhanced graph contains four types of nodes and five types of edges where mention nodes, mention-utterance (MU) edges and coreference-chain (CC) edges are proposed based on DialogRE^{C+}. 1) Mention nodes are composed of elements in the coreference chain. 2) The MU edge is established to connect a mention node and an utterance node, if the mention is occured in the utterance. 3) In order to establish the interaction among mentions, we fully connect the mention nodes in the coreference chain using CC edges.

[6] We do not compare with D^{2}G [4] as it uses many other structural features than coreference information (e.g., dependency tree and dialogue answering links), which may cause unfair comparisons.



(a) Graph in TUCORE-GCN^{C+} (b) Graph in REDialog^{C+}

(c) Graph in GAIN^{C+} (d) Graph in HGAT^{C+}

Fig. 3. The coreference-enhanced graph in backbone models for the example in Fig. 1. Arguments are S2 “Pheebs” and her brother “Frank”. The explanations of nodes and edges can be found in the paper of backbone models, except for the red ones, which are only used in this paper because they depend on coreference information. The gray backgrounds denote the coreference clusters containing argument entities and their mentions.

REDialog^{C+} [29] receives an dialogue with coreference chains as input. First, we use speaker embeddings to represent the speakers of sentences and concatenate them with word embeddings obtained from BERT [3]. Then, we construct a fully-connected coreference-enhanced graph using the feature representation. Finally, the argument nodes after GCN are used for classification.

As shown in Fig. 3.(b), the graph contains three types of nodes. 1) Each mention node corresponds to one mention in the coreference chain of the arguments. The coreference chains of REDialog only contain rule-based pronouns “I” and “you”, while our coreference chains also contain rich coreference information such as third person pronouns and possessor pronouns. 2) Argument nodes are the average representations of the corresponding mention nodes. 3) MDP indicates a set of shortest dependency paths for the target entity mentions, and tokens in the MDP are extracted as MDP nodes.

GAIN^{C+} constructs two graphs based on [28]. As shown in Fig. 3.(c), the coreference-enhanced graph contains dialogue node which aims to model the overall dialogue information, and mention nodes denoting each mention of arguments. The introduction of coreference chains enriches the mention nodes in the graph, helping to capture the arguments features distributed in dialogue. The entity-level graph aggregates mentions for the same entities in the former graph.

HGAT^{C+} [2] constructs a graph that contains five types of nodes: argument nodes, utterance nodes, speaker nodes, type nodes and word nodes, where speaker nodes represent each unique speaker in the input dialogue, type nodes denote the word types like PERSON and LOCATION and word nodes denote the vocabulary of dialogue, as shown in Fig. 3.(d). It also contains eight types of edges, utterance-word/argument/speaker edge, type-word/argument edge and coreference-word(CW)/utterance/(CU)/speaker(CS) edge. Each word or argument is connected with the utterances that contain the word or argument, and it is also connected with corresponding types. Each speaker is connected with the utterances uttered by the speaker.

Due to the introduction of coreference chains, three types of edges have been added to the graph. The CW edges connect the argument and the words in its coreference chain. The CS edges will connect the argument and the same speaker if the argument is a speaker. We have also added CU edges to connect the argument and the utterances where the mentions of this argument occur.

4.2 Coreference Resolution Model

We use E2E-coref [12], the first end-to-end coreference resolution model, to explore the impact of automatic extraction of coreference chains on the DER task. We train the E2E-coref model using the coreference data from the CoNLL-2012 task [18] and our DialogRE^{C+} dataset, respectively. It formulate the coreference resolution task as a set of decisions for every possible span in the document. For the i-th span, the task involves predicting its antecedent y_i based on the coreference score. The set of possible assignments for each y_i is $\{\epsilon, 1, \ldots, i-1\}$, where ϵ is a dummy antecedent. First, we obtain the span representations g through pretrained word embeddings and BiLSTM. Then, the mention score s_m and antecedent score s_a are calculated based on span representations:

$$s_m(i) = w_m \cdot FFNN_m(g_i) \tag{1}$$

$$s_a(i,j) = w_a \cdot FFNN_a([g_i, g_j, g_i \circ g_j, \phi(i,j)]) \tag{2}$$

where w_m and w_a are trainable parameters, $FFNN$ denotes a feed-forward neural network, $\cdot$ denotes the dot product, $\circ$ denotes element-wise multiplication, and $\phi(i,j)$ denotes the feature vector from the metadata. The coreference score $s(i,j)$ between span i and span j can be obtained from s_m and s_a:

$$s(i,j) = s_m(i) + s_m(j) + s_a(i,j) \tag{3}$$

For more technical details, we recommend referring to the original paper [12].

5 Experiments and Analyses

5.1 Experiment Setup

We conduct experiments on DialogRE and DialogRE^{C+} and calculate F1 scores as evaluation metrics. For all the BERT representations, we use BERT-base-uncased model (768d) as the encoder. For GAIN-GloVe, we use the GloVe embedding (100d) and BiLSTM (256d) as word embedding and encoder. For HGAT,

Table 2. Model performance comparison. We run each experiment five times and report the average F1 along with standard deviation (σ).

Model	Dev				Test			
	ori(σ)	+cof(σ)	+ret(σ)	+zs(σ)	**ori(σ)**	**+cof(σ)**	**+ret(σ)**	**+zs(σ)**
TUCORE	66.8(0.7)	68.8(0.3)	67.9(0.6)	67.5(0.5)	65.5(0.4)	67.8(0.4)	67.0(0.9)	65.8(1.0)
REDialog	63.0(0.4)	65.6(1.1)	63.8(0.8)	61.2(0.3)	62.8(1.5)	65.2(1.1)	62.5(0.6)	59.8(0.3)
GAIN$_{BERT}$	66.1(0.4)	68.4(0.4)	65.1(0.3)	63.5(0.3)	63.6(0.8)	67.2(0.6)	63.7(0.3)	61.6(0.7)
GAIN$_{GloVe}$	52.3(0.9)	57.2(1.0)	55.3(0.5)	53.3(0.8)	55.8(0.3)	58.2(0.4)	56.3(0.5)	55.8(0.4)
HGAT	56.7(0.3)	59.0(0.3)	57.9(0.6)	57.2(0.3)	59.4(0.7)	54.7(0.8)	57.9(1.0)	53.1(0.7)
Ave	-	+2.8	+1.0	-0.4	-	+3.2	+0.6	-1.0

we use the GloVe embedding (300d) and BiLSTM (128d) as word embedding and encoder. We set the learning rates of TUCORE-GCN, REDialog, GAIN-BERT GAIN-GloVe, and HGAT to 3e-5, 1e-5, 1e-3, 1e-3, and 1e-4, respectively.

5.2 Main Results

Table 2 reports the evaluation results[7]. For each of the base models, we compare the performances of the original model (denoted as **ori**), the model with manually annotated coreference information (**+cof**), the model with the coreference information generated from a pre-trained coreference resolution model (**+ret**), the model with the coreference information parsed from an off-the-shelf coreference resolution tool [12] (namely zero-shot, **+zs**). From Table 2, we have the following observations: 1) The annotated coreference chains introduce stable F1 improvements for all base models, with an average improvement of 2.8 and 3.2 on dev set and test set respectively. This validates the effectiveness of coreference chains on graph-based models in relational reasoning. 2) The retrained coreference chains achieve a lower improvement effect than the annotated ones. This demonstrates the practicality of coreference chains in relation extraction. 3) The zero-shot coreference chains do not show any improvement effect on DRE, indicating the necessity of annotating DialogRE^{C+} dataset.

Table 3. Ablation studies of different edge types.

Model	dev	test
TUCORE-GCN^{c+}	**68.8**	**67.8**
w/o CC edge	67.4	66.8
w/o MU edge	67.9	67.5
w/o CC and MU edge	66.8	65.5

Table 4. Performance comparison of inter- and intra-utterance relation extraction.

Method	dev		test	
	Inter	Intra	Inter	Intra
TUCORE-GCN^{c+}	65.9	65.4	66.3	63.8
TUCORE-GCN	65.3	63.7	65.2	63.5

[7] We remove the rule-based person references from original model of REDialog.

5.3 Effect of Coreference-Enhanced Graphs

We conducted ablation experiments based on TUCORE-GCN^{c+}. We remove the coreference-chain and mention-utterance edges respectively and simultaneously. The coreference-chain edges are removed to judge whether the coreference information has the ability to learn from each other, the mention-utterance edges are removed to prove the effect of location information of conference chain. The graph structure of TUCORE-GCN is formed by removing both the coreference-chain edges and mention-utterance edges. Table 3 shows the result of the above three experiments, the names of edges are capitalized, for example, coreference-chain edge is represented as CC edge. The result shows that the use of coreference chains to enrich the dialogue graph is beneficial for relationship extraction.

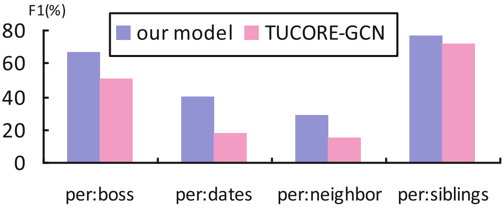


Fig. 4. Effect analysis for partial relations.

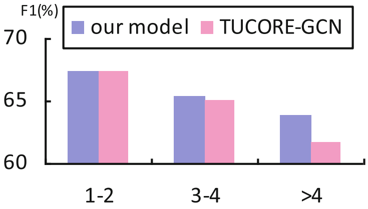


Fig. 5. Effect analysis for different speaker numbers.

5.4 Effect Analysis for Partial Relationships

In Fig. 4, we compare the performances of TUCORE-GCN^{c+} and TUCORE-GCN in extracting partial relationships. Obviously, in those relationships, such as per:boss and per:neighbor, the prediction effect of the model is significantly improved. The reason may be that in the communication, the speaker will add the possessive pronouns of sequential adjectives when referring to this kind of relationship, such as 'your, his, her', when joining the labeled pronouns, the model performs well in the prediction process.

5.5 Impact of Coreference on Speaker Numbers

In the Fig. 5, we compare the performance of TUCORE-GCN^{c+} and TUCORE-GCN in the dialogues with different speaker numbers. When the number of speakers is greater than 4, the effect of the model is improved after adding coreferential information. This is because when there are a large number of speaker, the dialogue is generally longer, and the prediction effect in the longer text can be improved by adding coreference information.

5.6 Impact of Coreference on Inter- And Intra-Utterance Relations

In this section, we make a comparative analysis for inter- and intra-utterance RE using TUCORE-GCN^{c+} and TUCORE-GCN. In Table 4, we can find that in the dev and test datasets, the F1 of inter-utterance relation extraction increased by 0.6 and 1.1, which shows that joining coreference information can promote cross-utterance relationship extraction effect, which is due to join mention node in different utterances and join different mention in the same utterance, thus the model can learn more information in different utterances more accurately, so as to improve the effect of inter-utterance relation extraction.

6 Conclusion and Future Work

In this paper, we annotate coreference information based on the DialogRE dataset, proposing the first dialogue relation extraction dataset with coreference chains. Based on this dataset, we build 4 coreference-enhanced graph-based models for DRE. By adding annotated coreference information into DRE models, the effects of the models exceed the baselines significantly. We also add automatically extracted coreferentce chains into DRE models. The coreferentce chains given by a retrained coreference resolution model help DRE models to some extent, while the ones given by an off-the-shelf coreference resolution model worsen the performances of DRE models, demonstrating the necessity of annotating coreferentce chains on the DRE task.

In the following research work, we hope to make contributions to the identification of dialogue coreference information based on the proposed DialogRE^{C+} dataset, use coreference chains to improve DRE and apply the experimental results to other dialogue-level tasks. We are also interested in joint research on the DRE task and coreference resolution task.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (No. 2022YFB3103602) and the National Natural Science Foundation of China (No. 62176187).

References

1. Cao, H., et al.: OneEE: A one-stage framework for fast overlapping and nested event extraction. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 1953–1964 (2022)
2. Chen, H., Hong, P., Han, W., Majumder, N., Poria, S.: Dialogue relation extraction with document-level heterogeneous graph attention networks. Cogn. Comput. **15**, 793–802 (2023)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL, pp. 4171–4186 (2019)

4. Fei, H., Li, J., Wu, S., Li, C., Ji, D., Li, F.: Global inference with explicit syntactic and discourse structures for dialogue-level relation extraction. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI, pp. 4082–4088 (2022)
5. Fei, H., Ren, Y., Ji, D.: Boundaries and edges rethinking: an end-to-end neural model for overlapping entity relation extraction. Inf. Process. Manage. **57**(6), 102311 (2020)
6. Fei, H., et al.: Lasuie: unifying information extraction with latent adaptive structure-aware generative language model. In: Proceedings of the Advances in Neural Information Processing Systems, NeurIPS 2022, pp. 15460–15475 (2022)
7. Fei, H., Zhang, M., Zhang, M., Chua, T.S.: Constructing code-mixed universal dependency forest for unbiased cross-lingual relation extraction. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 9395–9408 (2023)
8. Fei, H., Zhang, Y., Ren, Y., Ji, D.: A span-graph neural model for overlapping entity relation extraction in biomedical texts. Bioinformatics **37**(11), 1581–1589 (2021)
9. Joshi, M., Levy, O., Zettlemoyer, L., Weld, D.S.: Bert for coreference resolution: Baselines and analysis. In: Proceedings of EMNLP-IJCNLP, pp. 5803–5808 (2019)
10. Kirstain, Y., Ram, O., Levy, O.: Coreference resolution without span representations. In: Proceedings of ACL-IJCNLP, pp. 14–19 (2021)
11. Lee, B., Choi, Y.S.: Graph based network with contextualized representations of turns in dialogue. In: Proceedings of EMNLP, pp. 443–455 (2021)
12. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. In: Proceedings of EMNLP, pp. 188–197 (2017)
13. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 10965–10973 (2022)
14. Li, J., Xu, K., Li, F., Fei, H., Ren, Y., Ji, D.: MRN: a locally and globally mention-based reasoning network for document-level relation extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1359–1370 (2021)
15. Loáiciga, S.: Anaphora resolution for machine translation (résolution d'anaphores et traitement des pronoms en traduction automatique à base de règles)[in french]. In: Proceedings of TALN, pp. 683–690 (2013)
16. Long, X., Niu, S., Li, Y.: Consistent inference for dialogue relation extraction. In: IJCAI, pp. 3885–3891 (2021)
17. Nan, G., Guo, Z., Sekulić, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: Proceedings of ACL, pp. 1546–1557 (2020)
18. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: Conll-2012 shared task: modeling multilingual unrestricted coreference in ontonotes. In: Joint conference on EMNLP and CoNLL-shared task, pp. 1–40 (2012)
19. Qiu, L., et al.: SocAoG: incremental graph parsing for social relation inference in dialogues. In: Proceedings of ACL and IJCNLP, pp. 658–670 (2021)
20. Steinberger, J., Poesio, M., Kabadjov, M.A., Ježek, K.: Two uses of anaphora resolution in summarization. Inf. Process. Manage. **43**(6), 1663–1680 (2007)
21. Strube, M., Müller, C.: A machine learning approach to pronoun resolution in spoken dialogue. In: Proceedings of ACL, pp. 168–175 (2003)
22. Tang, H., et al.: Hin: hierarchical inference network for document-level relation extraction. In: Proceedings of PAKDD 2020, pp. 197–209. Springer (2020)

23. Wang, F., et al.: Entity-centered cross-document relation extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 9871–9881 (2022)
24. Wu, M., Moosavi, N., Roth, D., Gurevych, I.: Coreference reasoning in machine reading comprehension. In: Proceedings of ACL-IJCNLP, vol. 1, pp. 5768–5781. Association for Computational Linguistics (2021)
25. Xue, Z., Li, R., Dai, Q., Jiang, Z.: Corefdre: Document-level relation extraction with coreference resolution. arXiv preprint arXiv:2202.10744 (2022)
26. Yao, Y., et al.: DocRED: a large-scale document-level relation extraction dataset. In: Proceedings of ACL, pp. 764–777 (2019)
27. Yu, D., Sun, K., Cardie, C., Yu, D.: Dialogue-based relation extraction. In: Proceedings of ACL, pp. 4927–4940 (2020)
28. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: Proceedings of EMNLP, pp. 1630–1640 (2020)
29. Zhou, M., Ji, D., Li, F.: Relation extraction in dialogues: a deep learning model based on the generality and specialty of dialogue text. IEEE/ACM TASLP **29**, 2015–2026 (2021)
30. Zhou, P., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of ACL, pp. 207–212 (2016)

A Bi-directional Multi-hop Inference Model for Joint Dialog Sentiment Classification and Act Recognition

Li Zheng, Fei Li(✉), Yuyang Chai, Chong Teng, and Donghong Ji

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
{zhengli,lifei_csnlp,yychai,tengchong,dhji}@whu.edu.cn

Abstract. The joint task of Dialog Sentiment Classification (DSC) and Act Recognition (DAR) aims to predict the sentiment label and act label for each utterance in a dialog simultaneously. However, current methods encode the dialog context in only one direction, which limits their ability to thoroughly comprehend the context. Moreover, these methods overlook the explicit correlations between sentiment and act labels, which leads to an insufficient ability to capture rich sentiment and act clues and hinders effective and accurate reasoning. To address these issues, we propose a Bi-directional Multi-hop Inference Model (BMIM) that leverages a feature selection network and a bi-directional multi-hop inference network to iteratively extract and integrate rich sentiment and act clues in a bi-directional manner. We also employ contrastive learning and dual learning to explicitly model the correlations of sentiment and act labels. Our experiments on two widely-used datasets show that BMIM outperforms state-of-the-art baselines by at least 2.6% on F1 score in DAR and 1.4% on F1 score in DSC. Additionally, Our proposed model not only improves the performance but also enhances the interpretability of the joint sentiment and act prediction task.

Keywords: Dialog sentiment classification · Act recognition · Contrastive learning · Dual learning · Bi-directional joint model

1 Introduction

Dialog Sentiment Classification (DSC) and Act Recognition (DAR) have attracted increasing attention in the field of dialog-based natural language understanding [2–4]. DSC aims to detect the emotion (e.g., negative) expressed by a speaker in each utterance of the dialog, while DAR seeks to assign a semantic label (e.g., question) to each utterance and characterize the speaker's intention. Recent studies demonstrate that these two tasks are closely relevant, and how to exploit the correlations between them and thoroughly understand the context are key factors. Thus nichetargeting models are proposed to jointly address these two tasks by utilizing their correlations [1,5,6].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 235–248, 2023.
https://doi.org/10.1007/978-3-031-44693-1_19

Table 1. A dialog snippet from the Mastodon dataset [1] for joint dialog sentiment classification and act recognition.

Speaker	Utterances	Act	Sentiment
A	u_1: There's no way to make a post visible to just your local tl and not federated tl	Statement	Negative
B	u_2: Correct ?	Question	Negative
B	u_3: I don't think there is	Answer	Negative
A	u_4: Thanks	Thanking	Positive
B	u_5: Didn't think so	Agreement	Negative

Despite promising performance, most prior approaches only encode the dialog context in one direction, i.e., the chronological order of the utterances [7,8]. However, such approaches neglect the subsequent utterances after the target utterance which also play important roles in sentiment classification and act recognition. As shown in Table 1, the sentiment expressed by u_2 "correct?" is obscure when considering only the dialog context before u_2. Nevertheless, if we check the dialog context after u_2, we can observe that the subsequent utterances u_3 and u_5 from the same speaker both tend to express negative sentiment. Therefore, it is easy to infer the sentiment label for u_2 as "Negative", which highlights the necessity of bi-directional inference.

Moreover, existing works only implicitly exploite the correlations between sentiment and act labels [1], or even disregard the correlations at all [8]. The lack of explicit modeling results in an insufficient ability to capture rich sentiment and act clues, which prevents effective and accurate reasoning. Intuitively, considering the tight associations between sentiments and acts, it is beneficial to explicitly model their correlations. For instance, the sentiment labels of most utterances in the dialog of Table 1 are negative. A model that solely considers the dialog context is apt to incorrectly predict the sentiment label of u_4 as "Negative". In contrast, if the model is capable of explicitly considering the correlations between sentiments and acts, it can deduce that the act label "Thanking" is more likely to be associated with a positive sentiment label. Hence, explicitly modeling the correlations is necessary for both interpretability and performance improvement of joint sentiment and act prediction.

In this paper, we propose a Bi-directional Multi-hop Inference Model (BMIM) to model the dialog context in a bi-directional manner and explicitly exploit the correlations of sentiment and act labels. Firstly, we design a feature selection network to capture sentiment-specific and act-specific features, as well as their interactions, while removing the effect of multi-task confounders. Next, we leverage a bi-directional multi-hop inference network to iteratively extract and integrate rich sentiment and act clues from front to back and vice versa, emulating human's reasoning and cognition. Then, we employ contrastive learning and dual learning to explicitly model the correlations of sentiments and acts, meanwhile increasing the interpretability of our model. Finally, we utilize two classifiers to predict the sentiment and act labels for each utterance based on the refined features from aforementioned modules.

To verify the effectiveness of our model, we conduct experiments on two widely-used datasets for DSC and DAR, namely Mastodon [1] and Dailydialog [9]. The experimental results show that our model significantly outperforms all state-of-the-art baselines by at least 2.6% on F1 score in DAR and 1.4% on F1 score in DSC. Additionally, we conduct extensive experiments to show that our model has decent interpretability, such as visualization of the correlations between sentiment and act labels, visualization of the correlations between sentiment and act distributions, calculation of casual effect scores [10]. In conclusion, the contributions of this paper can be summarized as follows:

- We propose a novel bi-directional multi-hop inference model to analyze sentiments and acts in dialogs by understanding dialog contexts based on the way of imitating human reasoning.
- We employ contrastive learning and dual learning to explicitly model the correlations between sentiments and acts, leading to reasonable interpretability.
- We conduct extensive experiments on two public benchmark datasets, pushing the state-of-the-art for sentiment and act analyses in dialog.

2 Related Work

2.1 Sentiment and Emotion Analyses in NLP

Sentiment analysis [11–14] has long been an important research topic in natural language processing (NLP), deriving many research directions such as aspect-based sentiment analysis (ABSA) [15], emotion detection [16] and emotion recognition in conversations (ERC) [17] and emotion-cause pair extract (ECPE) [18–20]. ABSA focuses on detecting the sentiment polarities of different aspects in the same sentence [21]. ECPE considers emotions and their associated causes. In this paper, we focus on a new scenario, where not only the speaker's sentiments but also their acts should be extracted from the utterances.

2.2 Sentiment Classification and Act Recognition

Dialog Sentiment Classification and Act Recognition are sentence-level sequence classification problems, and it has been found that they are correlated [1,8]. Several joint models are proposed to enhance their mutual interactions, providing a more comprehensive understanding of the speaker's intentions. Li et al. [22] propose a context-aware dynamic convolution network (CDCN) to model the local contexts of utterances, but not consider the interactions between tasks. Qin et al. [7] propose a co-interactive graph attention network (Co-GAT) to consider both contextual information and mutual interaction information, but they ignore the role of label information. Xing et al. [6] propose a dual-task temporal relational recurrent reasoning network (DARER) to achieve prediction-level interactions and estimate label distributions. However, the aforementioned methods only model the dialog context in a one-way manner, disregarding the explicit correlations of labels between tasks and lacking interpretability. In contrast, we

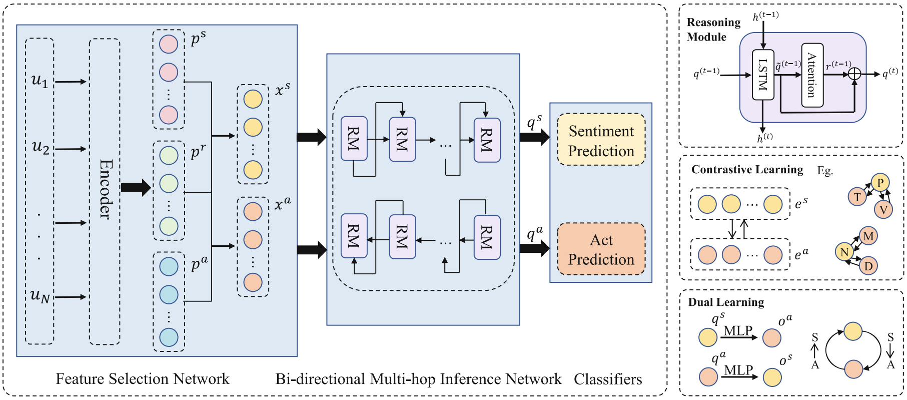


Fig. 1. The overview of our model. p^s, p^a and p^r denote sentiment, act and shared features. RM means Reasoning module (cf. Sect. 3.3). Contrastive learning is applied on sentiment and act label embeddings e^s and e^a to pull the ones frequently occurring in the same utterance close. P and N represent sentiment labels, positive and negative, while T, V, M and D represent act labels, namely thanking, explicit performative, sympathy and disagreement. When using dual learning, sentiment and act predictions are no longer parallel, they are conducted as pipeline to mimic causal inference. (cf. Sect. 3.4 for details).

propose a bi-directional multi-hop inference model to bi-directional capture the dialog context and explicitly model the correlations of sentiment and act labels using contrastive learning and dual learning.

2.3 Contrastive Learning and Dual Learning

Contrastive learning is a label-efficient representation learning mechanism that enhances the proximity of positive samples and increases the distance between negative samples, which exhibits advantages in various domains [23,24]. Besides, dual learning has been widely adopted in various tasks [25–27], including Question Answering/Generation [28] and Automatic Speech Recognition/Text-to-Speech [26]. The primal task and the dual task form a closed loop, generating informative feedback signals that can benefit both tasks. In this work, we present two approaches to enhance the interpretability of our model: contrastive learning and dual learning. Contrastive learning integrates logical dependencies between labels into dialog comprehension, whereas dual learning promotes mutual learning between the two tasks by incorporating logical dependencies.

3 Methodology

In this paper, we propose a bi-directional multi-hop inference model to extract features from the dialog context in both directions and explicitly exploit the correlations between sentiment and act labels. The architecture of our model is illustrated in Fig. 1 and comprises four components. First, we design a feature selection network to extract specific sentiment and act features and purify the shared features between them. Then, we propose a bi-directional multi-hop inference network to encode the dialog context from front to end and vice versa. Afterward, we explicitly model the correlations between sentiment and act labels employing contrastive learning and dual learning, respectively. Finally, we utilize two classifiers for dialog sentiment and act prediction.

3.1 Task Definition

Let $U = \{u_1, u_2, ..., u_N\}$ be a dialog, where N is the number of utterances. Our goal is to map utterance sequences $(u_1, u_2, ..., u_N)$ to the corresponding utterance sequence sentiment labels $(y_1^s, y_2^s, ..., y_N^s)$ and act labels $(y_1^a, y_2^a, ..., y_N^a)$.

3.2 Utterance Encoding

Following Xing et al. [6], we also apply BiLSTM [29] as the encoder to yield initial utterance representations $U = \{u_1, u_2, . . . , u_N\}$. Next, we leverage a feature selection network (FSN) [30] to extract task-specific and interactive features and remove the influence of redundant information. FSN divides neurons into three partitions (sentiment, act, and shared) at each time step, generating task-specific features by selecting and combining these partitions and filtering out irrelevant information. Specifically, at the i-th time step, we generate two task-related gates:

$$s_i = Cummax(Linear([u_i; h_{i-1}])), a_i = 1 - Cummax(Linear([u_i; h_{i-1}])) \quad (1)$$

where $Cummax(\cdot)$ denotes the cumulative maximum operation that performs as a binary gate, $Linear(\cdot)$ refers to a linear transformation, and h_{i-1} represents the hidden state of the $(i-1)$-th utterance. Each gate corresponds to a specific task and divides the utterance representations into two segments according to their relevance to the respective task. With the joint efforts of the two gates s_i and a_i, the utterance representations can be divided into three partitions: the sentiment partition p_i^s, the act partition p_i^a and the shared partition p_i^r. Next, we concatenate the sentiment and act partition separately with the shared partition to gain task-specific sentiment and act feature representations x_i^s and x_i^a:

$$x_i^s = tanh(p_i^s) + tanh(p_i^r), \quad x_i^a = tanh(p_i^a) + tanh(p_i^r) \quad (2)$$

3.3 Bi-directional Multi-hop Inference Network

To imitate human reasoning and mine the internal dependencies of utterances to thoroughly understand the context, we propose a bi-directional multi-hop inference network. Concretely, in the t-th turn, we adopt the LSTM network to learn intrinsic logical order and integrate contextual clues in the working memory, formulated as:

$$q_i^{(0)} = W_q x_i + b_q \quad (3)$$

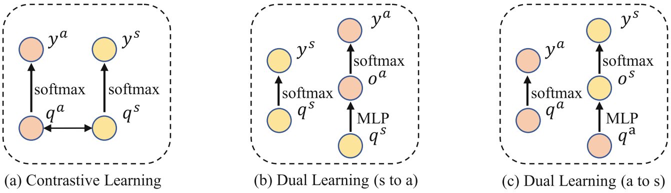


Fig. 2. Three architectures of the output layer of our model. Architecture (a) applies contrastive learning on sentiment and act representations, so sentiment and act distributions can be predicted in parallel. Architecture (b) and (c) apply dual learning where act or sentiment distributions are predicted grounded on the features given by the sentiment or act task.

$$\tilde{q}_i^{(t-1)}, h_i^{(t)} = \overrightarrow{LSTM}(q_i^{(t-1)}, h_i^{(t-1)}) \tag{4}$$

where W_q and b_q are learnable parameters. x_i can be either x_i^s or x_i^a. $h_i^{(t)}$ is the working memory, which stores and updates the previous memory $h_i^{(t-1)}$ and guides the next turn of clue extraction. t denotes the number of inference steps.

In order to mimic human retrieval and reasoning processes, we utilize an attention mechanism to mine relevant contextual clues:

$$e_{ij}^{(t-1)} = f(x_j, \tilde{q}_i^{(t-1)}),\ \alpha_{ij}^{(t-1)} = \frac{exp(e_{ij}^{(t-1)})}{\sum_{j=1}^{N} exp(e_{ij}^{(t-1)})},\ r_i^{(t-1)} = \sum_{j=1}^{N} \alpha_{ij}^{(t-1)} x_j \tag{5}$$

where f is a dot product function. Then, we concatenate the output of inference process $\tilde{q}_i^{(t-1)}$ with the resulting attention readout $r_i^{(t-1)}$ to form the next-turn queries $q_{f\to b}^{(t)}$ and $q_{b\to f}^{(t)}$ from front to back and vice versa to explore contextual dependencies:

$$q_{f\to b}^{(t)} = [\tilde{q}_{f\to b}^{(t-1)}; r_{f\to b}^{(t-1)}], \quad q_{b\to f}^{(t)} = [\tilde{q}_{b\to f}^{(t-1)}; r_{b\to f}^{(t-1)}] \tag{6}$$

Based on the above output vectors, the final sentiment and act representations q^s and q^a with rich contextual clues can be defined as a concatenation of both vectors:

$$q^s = [q_{f\to b}^s; q_{b\to f}^s], \quad q^a = [q_{f\to b}^a; q_{b\to f}^a] \tag{7}$$

3.4 Contrastive Learning and Dual Learning

Contrastive Learning. To model the correlations between labels, bringing related sentiment and action labels closer while pushing unrelated labels away, we employ contrastive learning [24]. Our contrastive loss function is defined as:

$$\mathcal{L}^{cl} = \sum_{i=1}^{l} \frac{-1}{|\mathcal{P}|} \sum_{p\in\mathcal{P}} \log \frac{\exp(e_i \cdot e_p/\tau)}{\sum_{p\in\mathcal{P}} \exp(e_i \cdot e_p/\tau) + \sum_{n\in\mathcal{N}} \exp(e_i \cdot e_n/\tau) + \varepsilon} \tag{8}$$

where l is the total category size of sentiment and act labels and e_i is the label embedding representation. The positive set $\mathcal{P}$ contains the indexes of co-existed labels with the label e_i in the training batch, while the negative set $\mathcal{N}$ contains the indexes of labels that never co-existed with e_i in the training set.

Dual Learning. We incorporate dual learning into our approach to facilitate mutual learning of the two tasks, allowing for a more comprehensive understanding and utilization of the logical dependencies between sentiment and act labels. Rather than solely considering inter-task duality, we also take into account the causalities of sentiments on acts and acts on sentiments to further improve the model performance. Specifically, we consider the impact of sentiment on act and predict the act score $o^a = MLP(q^s)$, based on a multi-layer perceptron and sentiment representation q^s. Then we employ a classifier to predict dialog act and sentiment for each utterance as $y^a = Softmax(o^a) = P(a|s;\theta_{s\to a})$. Similarly, we can perform such process reversely and model the impact of act on sentiment, formulated as: $y^s = Softmax(MLP(q^s))$. To enforce the dual learning process, we have the following loss:

$$\mathcal{L}^{dl} = (log\hat{P}(s) + logP(a|s;\theta_{s\to a}) - log\hat{P}(a) - logP(s|a;\theta_{a\to s}))^2 \tag{9}$$

where $y^a = P(a|s;\theta_{s\to a})$ and $y^s = P(s|a;\theta_{a\to s})$. Note that the true marginal distribution of data $P(s)$ and $P(a)$ are often intractable, so here we substitute them with the approximated empirical marginal distribution $\hat{P}(a)$ and $\hat{P}(s)$ [25].

3.5 Joint Training

The training loss is defined as the cross-entropy loss between predicted label distributions and ground-truth label distributions in the training set:

$$\mathcal{L}^s = -\sum_{i=1}^{N}\sum_{j=1}^{N_s} \hat{y}_{ij}^s log(y_{ij}^s), \quad \mathcal{L}^a = -\sum_{i=1}^{N}\sum_{j=1}^{N_a} \hat{y}_{ij}^a log(y_{ij}^a) \tag{10}$$

where y_{ij}^s, y_{ij}^a, $\hat{y}_{ij}^s$ and $\hat{y}_{ij}^a$ are the predicted and gold sentiment and act distributions for the i-th utterance. N_s and N_a are the numbers of sentiment and act labels. The aforementioned 4 losses can be combined and applied on the three architectures in Fig. 2, where Eq. 11, 12, 13 corresponding to Architecture (a), (b) and (c) respectively:

$$\mathcal{L}^c = \mathcal{L}^s + \mathcal{L}^a + \mathcal{L}^{cl} \tag{11}$$

$$\mathcal{L}^{d,s\to a} = \mathcal{L}^s + \mathcal{L}^a + \mathcal{L}^{dl}_{s\to a} \tag{12}$$

$$\mathcal{L}^{d,a\to s} = \mathcal{L}^s + \mathcal{L}^a + \mathcal{L}^{dl}_{a\to s} \tag{13}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We assess the efficacy of our model on two publicly dialog datasets, Mastodon [1] and Dailydialog [9]. The Mastodon dataset consists 269 dialogs

Table 2. Comparison of our model with baselines on Mastodon and Dailydialog datasets. BMIM^{c}, BMIM$^{d,s\rightarrow a}$ and BMIM$^{d,a\rightarrow s}$correspond to architecture (a), (b) and (c) in Fig. 2, respectively.

Models	Mastodon						DailyDialog					
	DSC			DAR			DSC			DAR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
JointDAS	36.1	41.6	37.6	55.6	51.9	53.2	35.4	28.8	31.2	76.2	74.5	75.1
IIIM	38.7	40.1	39.4	56.3	52.2	54.3	38.9	28.5	33.0	76.5	74.9	75.7
DCR-Net	43.2	47.3	45.1	60.3	56.9	58.6	56.0	40.1	45.4	79.1	79.0	79.1
BCDCN	38.2	62.0	45.9	57.3	61.7	59.4	55.2	45.7	48.6	80.0	80.6	80.3
Co-GAT	44.0	53.2	48.1	60.4	60.6	60.5	65.9	45.3	51.0	81.0	78.1	79.4
TSCL	46.1	58.7	51.6	61.2	61.6	60.8	56.6	49.2	51.9	78.8	79.8	79.3
DARER	56.0	63.3	59.6	65.1	61.9	63.4	60.0	49.5	53.4	81.4	80.8	81.1
BMIM^{c}	58.2	64.0	**61.0**	68.2	62.7	**65.3**	60.0	49.7	54.3	83.9	83.2	83.5
BMIM$^{d,s\rightarrow a}$	58.0	62.2	60.0	67.2	61.8	64.4	59.9	49.6	54.3	83.9	83.2	**83.7**
BMIM$^{d,a\rightarrow s}$	58.7	62.4	60.5	67.1	61.8	64.3	60.0	49.7	**54.4**	83.8	83.1	83.5

with 1075 utterances allocated to training and 266 dialogs with 1075 utterances reserved for testing. It encompasses 3 sentiment categories and 15 act categories. To ensure consistency with Cerisara et al. [1], we follow the same partition scheme.

Evaluation Metrics. Following previous works [1,6], we exclude the neutral sentiment label in DSC. For DAR, we employ the average of the F1 scores weighted by the prevalence of each dialog act on Mastodon. While on DailyDialog, we adopt the macro-averaged Precision (P), Recall (R) and F1 as the major metrics to measure the effectiveness of our model in both tasks.

4.2 Baseline Systems

To verify the effectiveness of the BMIM, we compare it with the following state-of-the-art baselines, which are categorized into three groups based on their modeling approaches. The first group solely focuses on modeling the context or interaction information, which includes JoinDAS [1] and IIIM [8]. The second group considers both the dialog context and implicit interaction information between tasks, and includes DCR-Net [31], BCDCN [22], Co-GAT [7] and TSCL [5]. The third group utilizes implicit label information and includes DARER [6]. Notably, all baselines only employ unidirectional modeling of the dialog context.

4.3 Overall Results

The experimental results for both DSC and DAR tasks are presented in Table 2. Our method demonstrates obvious advantages over other state-of-the-art baselines for both tasks. For instance, on Mastodon, our model achieves the best performance when employing contrastive learning, surpassing the best baseline (DARER) with an absolute improvement of 1.4% F1 score on the DSC task and 1.9% F1 score on the DAR task. Similarly, on DailyDialog, dual learning with the sentiment-to-act approach works best for the DAR task, outperforming the best baseline (DARER) by 2.6% F1 score. When using dual learning with the act-to-sentiment approach, the performance on DSC tasks is the best, with a 1.0% higher F1 score than the best baseline (DARER). We attribute the performance improvement to three aspects: (1) Our framework implements bidirectional modeling of dialog contexts, which infers context from a perspective that aligns with human thinking and obtains contextual representations with richer sentiment and act clues. (2) We design contrastive learning and dual learning to explicitly capture the correlations between sentiment and act labels, effectively utilizing label information. (3) Our feature selection network filters out task-independent information and optimally applies task-related one to promote performance.

4.4 Ablation Study

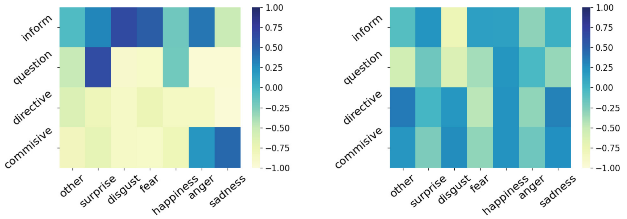
We conduct ablation experiments to assess the contribution of each component in our model. As depicted in Table 3, we observe that no variants can compete with the complete model, implying the indispensability of each component for the task. Specifically, the F1 score decreases most heavily without the bi-directional multi-hop inference network, which indicates that it has a significant effect on modeling the complex structural information of dialog. Furthermore, we investigate the necessity and effectiveness of contrastive learning and dual learning by removing these two modules. The sharp drops of results demonstrate that either contrastive learning or dual learning plays an important role in capturing explicit correlations between labels. Besides, removing the feature selection nerwork results in a distinct performance decline. This finding implies that efficient utilization of task-related information is able to enhance our model performance.

Table 3. Results of the ablation study on F1 scores.

Variants	Mastodon		DailyDialog	
	DSC	DAR	DSC	DAR
BMIM^c	**61.0**	**65.3**	**54.3**	**83.5**
w/o BMIN	58.2	63.6	51.8	81.7
w/o CL or DL	60.2	64.3	53.6	82.7
w/o FSN	59.5	64.1	52.9	82.5

4.5 Visualizing the Effectiveness of Contrastive Learning

We perform a visualization study in Fig. 3 to investigate the effectiveness of contrastive learning and demonstrate that it assists our model in learning the correlations between sentiment and act labels. Figure 3(a) illustrates that when using contrastive learning, each type of sentiment label has strongly and weakly correlated act labels. Strongly correlated labels are placed closer together, while weakly correlated labels are farther apart. In contrast, as depicted in Fig. 3(b), without contrastive learning, the correlations between labels is not significant, which limites the ability to effectively utilize the information between labels for



(a) with contrastive learning

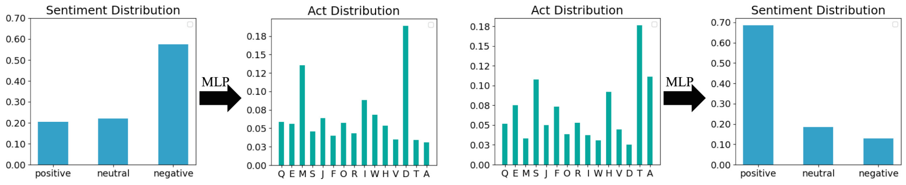
(b) without contrastive learning

Fig. 3. Visualization of the correlations between sentiment and act labels.

modeling purposes. These observations reveal the explicit correlations between sentiment and act labels and enhance the interpretability of our model. It also proves that our contrastive learning can proficiently model and utilize the correlations between labels to facilitate model inference.

4.6 Visualizing the Interpretability of Dual Learning

With the attempt to better understand how dual learning exploits explicit correlations between labels, we select two examples corresponding to the architecture (b) and (c) in Fig. 2, and visualize the distributions of sentiment and act labels in these examples. The results in Fig. 4 indicate that when the distribution value of the "negative" sentiment label reaches its maximum, the model is more likely to predict act labels such as "disagreement" (D) or "symmetry" (M). Similarly, larger distributions of act labels such as "thinking" (T), "agreement" (A), and "suggestion" (S) make it easier for the model to predict a "positive" sentiment label. In summary, dual learning effectively utilizes explicit correlations between labels, enhancing the model performance. This approach also improves the interpretability of such correlations, which aligns with human cognition.



(a) sentiment to act

(b) act to sentiment

Fig. 4. Visualization of the correlations between sentiment and act distributions.

4.7 Evaluating Causal Effect Scores of Our Model

To establish causal relationships between sentiment and act labels, which aligns with human prior knowledge, we estimate causal effects [10] between labels and present some results in Table 4. The act labels "Thanking" and "Greeting" have causal effects on "Positive" sentiment with scores of 0.32 and 0.28, respectively, and no impact on other sentiment types. In addition, the causal relationship between sentiment and act labels is not one-to-one. Rows 4–6 of Table 4 reveal that all act labels have significant causal relationships with two sentiment labels. Our quantitative analysis of the causal relationship between labels enhances the interpretability of our model and emphasizes the significance of explicitly modeling label information to improve performance.

Table 4. The estimated causal effects of sentiment and act labels on Mastodon. A higher score indicates more causal relationship. "–" denotes that the estimated effect is below 0.001.

	Positive	Neutral	Negative
Thanking	0.32	–	–
Greeting	0.28	–	–
Sympathy	–	0.14	0.26
Agreement	0.34	–	0.32
Disagreement	–	0.07	0.34

Table 5. Results based on different pre-trained language models.

Models	DSC			DAR		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
BERT + Linear	64.6	66.5	65.5	72.5	70.6	71.6
BERT + DARER	65.5	67.3	66.4	73.1	71.3	72.2
BERT + BMIM	67.9	70.3	**69.0**	76.6	73.9	**75.2**
RoBERTa + Linear	60.0	64.6	62.2	69.7	67.0	68.4
RoBERTa + DARER	60.7	65.3	62.9	70.0	67.9	68.9
RoBERTa + BMIM	62.8	67.7	**65.1**	74.2	71.4	**72.7**
XLNet + Linear	64.9	66.4	65.6	70.8	69.1	69.9
XLNet + DARER	67.3	68.4	67.8	71.9	69.5	70.7
XLNet + BMIM	68.9	70.3	**69.6**	74.1	73.1	**73.6**

4.8 Performances Using Different Pre-trained Language Models

Following Xing et al. [6], we also explore three pre-trained models, BERT, RoBERTa and XLNet in our framework. In this study, we replace the BiLSTM utterance encoder with these pre-trained models, while retaining the other components. We compare our approach with DARER, a competitive baseline, under different encoders and presented the results in Table 5. A single pre-trained encoder yields promising results, which highlights the excellent language comprehension abilities of pre-trained models. Moreover, our model consistently outperforms DARER on both tasks, irrespective of whether pre-trained models are used or not. Notably, our model wins DARER over 3.0% F1 score on DAR task when using BERT. Additionally, the performance gaps are further enlarged when using only a linear classifier on the pre-trained model. These outcomes indicate that our method can well model the dialog context bidirectionally while also exploiting the explicit correlations between labels.

5 Conclusion

In this work, we propose a Bi-directional Multi-hop Inference Model to tackle the joint task of DSC and DAR. BMIM leverages a feature selection network, a bi-directional multi-hop inference network, as well as contrastive learning and dual learning to iteratively extract and integrate abundant sentiment and act clues in a bi-directional manner, explicitly model the correlations of sentiment and act labels. Experimental results on two datasets demonstrate the effectiveness of our model, achieving state-of-the-art performance. Extensive analysis further confirms that our approach can proficiently comprehend the dialog context in both directions, and exploit the correlations between sentiment and act labels for better performance and interpretability.

Acknowledgment. This work is supported by the National Key Research and Development Program of China (No. 2022YFB3103602) and the National Natural Science Foundation of China (No. 62176187).

References

1. Cerisara, C., Jafaritazehjani, S., Oluokun, A., Le, H.T.: Multi-task dialog act and sentiment recognition on mastodon. In: Proceedings of the COLING, pp. 745–754 (2018)
2. Li, B., et al.: Revisiting conversation discourse for dialogue disentanglement. CoRR abs/2306.03975 (2023)
3. Fei, H., Li, J., Wu, S., Li, C., Ji, D., Li, F.: Global inference with explicit syntactic and discourse structures for dialogue-level relation extraction. In: Proceedings of the IJCAI, pp. 4082–4088 (2022)
4. Fei, H., Wu, S., Zhang, M., Ren, Y., Ji, D.: Conversational semantic role labeling with predicate-oriented latent graph. In: Proceedings of the IJCAI, pp. 4114–4120 (2022)
5. Xu, Y., Yao, E., Liu, C., Liu, Q., Xu, M.: A novel ensemble model with two-stage learning for joint dialog act recognition and sentiment classification. Pattern Recognit. Lett. **165**, 77–83 (2023)
6. Xing, B., Tsang, I.W.: DARER: dual-task temporal relational recurrent reasoning network for joint dialog sentiment classification and act recognition. In: Proceedings of the ACL, pp. 3611–3621 (2022)

7. Qin, L., Li, Z., Che, W., Ni, M., Liu, T.: Co-GAT: a co-interactive graph attention network for joint dialog act recognition and sentiment classification. In: Proceedings of the AAAI, pp. 13709–13717 (2021)
8. Kim, M., Kim, H.: Integrated neural network model for identifying speech acts, predicators, and sentiments of dialogue utterances. Pattern Recognit. Lett. **101**, 1–5 (2018)
9. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: a manually labelled multi-turn dialogue dataset. In: Proceedings of the IJCNLP, pp. 986–995 (2017)
10. Chen, W., Tian, J., Xiao, L., He, H., Jin, Y.: Exploring logically dependent multi-task learning with causal inference. In: Proceedings of the EMNLP, pp. 2213–2225 (2020)
11. Wu, S., et al.: Mastering the explicit opinion-role interaction: syntax-aided neural transition system for unified opinion role labeling. In: Proceedings of the AAAI, pp. 11513–11521 (2022)
12. Shi, W., Li, F., Li, J., Fei, H., Ji, D.: Effective token graph modeling using a novel labeling strategy for structured sentiment analysis. In: Proceedings of the ACL, pp. 4232–4241 (2022)
13. Fei, H., Chua, T., Li, C., Ji, D., Zhang, M., Ren, Y.: On the robustness of aspect-based sentiment analysis: rethinking model, data, and training. ACM Trans. Inf. Syst. **41**(2), 50:1–50:32 (2023)
14. Fei, H., Li, B., Liu, Q., Bing, L., Li, F., Chua, T.S.: Reasoning implicit sentiment with chain-of-thought prompting. In: Proceedings of the ACL, pp. 1171–1182 (2023)
15. Liang, B., Su, H., Gui, L., Cambria, E., Xu, R.: Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. Knowl.-Based Syst. **235**, 107643 (2022)
16. Fei, H., Zhang, Y., Ren, Y., Ji, D.: Latent emotion memory for multi-label emotion classification. In: Proceedings of the AAAI, pp. 7692–7699 (2020)
17. Li, B., et al.: DiaASQ: a benchmark of conversational aspect-based sentiment quadruple analysis. In: Findings of the ACL 2023, pp. 13449–13467 (2023)
18. Xia, R., Ding, Z.: Emotion-cause pair extraction: a new task to emotion analysis in texts. In: Proceedings of the ACL, pp. 1003–1012 (2019)
19. Chen, S., et al.: Joint alignment of multi-task feature and label spaces for emotion cause pair extraction. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 6955–6965 (2022)
20. Zheng, L., et al.: ECQED: emotion-cause quadruple extraction in dialogs. CoRR abs/2306.03969 (2023)
21. Fei, H., Li, F., Li, C., Wu, S., Li, J., Ji, D.: Inheriting the wisdom of predecessors: a multiplex cascade framework for unified aspect-based sentiment analysis. In: Proceedings of the IJCAI, pp. 4096–4103 (2022)
22. Li, J., Fei, H., Ji, D.: Modeling local contexts for joint dialogue act recognition and sentiment classification with bi-channel dynamic convolutions. In: Proceedings of the COLING, pp. 616–626 (2020)
23. Huang, C., Zhang, Z., Fei, H., Liao, L.: Conversation disentanglement with bi-level contrastive learning. In: Findings of the EMNLP, pp. 2985–2996 (2022)
24. Chai, Y., et al.: Prompt-based generative multi-label emotion prediction with label contrastive learning. In: Lu, W., Huang, S., Hong, Y., Zhou, X. (eds.) NLPCC 2022. LNCS, vol. 13551, pp. 551–563. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17120-8_43
25. Su, S., Huang, C., Chen, Y.: Dual supervised learning for natural language understanding and generation. In: Proceedings of the ACL, pp. 5472–5477 (2019)

26. Cao, R., Zhu, S., Liu, C., Li, J., Yu, K.: Semantic parsing with dual learning. In: Proceedings of the ACL, pp. 51–64 (2019)
27. Fei, H., Li, C., Ji, D., Li, F.: Mutual disentanglement learning for joint fine-grained sentiment classification and controllable text generation. In: Proceedings of the ACM SIGIR, pp. 1555–1565 (2022)
28. Fei, H., Wu, S., Ren, Y., Zhang, M.: Matching structure for dual learning. In: Proceedings of the ICML, pp. 6373–6391 (2022)
29. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
30. Yan, Z., Zhang, C., Fu, J., Zhang, Q., Wei, Z.: A partition filter network for joint entity and relation extraction. In: Proceedings of the EMNLP, pp. 185–197 (2021)
31. Qin, L., Che, W., Li, Y., Ni, M., Liu, T.: DCR-net: a deep co-interactive relation network for joint dialog act recognition and sentiment classification. In: Proceedings of the AAAI, pp. 8665–8672 (2020)

Positive-Guided Knowledge Distillation for Document-Level Relation Extraction with Noisy Labeled Data

Daojian Zeng[1,2], Jianling Zhu[1], Lincheng Jiang[3], and Jianhua Dai[1](✉)

[1] Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, China
{zengdj,zhujl,jhdai}@hunnu.edu.cn

[2] Institute of AI and Targeted International Communication, Hunan Normal University, Changsha, China

[3] College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, China
linchengjiang@nudt.edu.cn

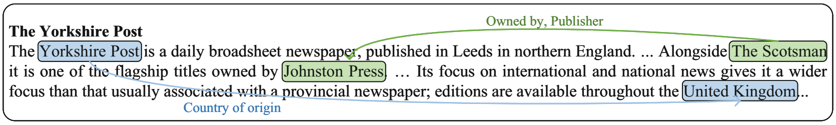
Abstract. Since one entity may have multiple mentions and relations between entities may stretch across multiple sentences in a document, the annotation of document-level relation extraction datasets becomes a challenging task. Many studies have identified that these datasets contain a large number of noisy labels, hindering performance improvement for the document-level relation extraction task. The previous and most straightforward method is denoising noisy labeled data from a data annotation perspective. However, this time-consuming approach is not suitable for large-scale datasets. In this paper, we propose a novel **P**ositive-**G**uided **K**nowledge **D**istillation (PGKD) model to address the noisy labeled data problem for document-level relation extraction. We design a new teacher-student architecture. The teacher model trained with only positive samples can partially supervise the student model. The positive-guided knowledge distillation algorithm transfers the clean positive-class patterns from the teacher model to the student model. In this way, the student model trained with all samples can efficiently prevent the interference of false negative samples. Extensive experiments on Mix-DocRED demonstrate that PGKD achieves state-of-the-art effectiveness for document-level relation extraction with noisy labeled data. Moreover, PGKD also surpasses other baselines even on the well-annotated Re-DocRED.

Keywords: Positive-guided knowledge distillation · Noisy labeled data · False negative samples · Document-level relation extraction

1 Introduction

Relation extraction (RE) comprises a primary branch of information extraction. It plays a crucial role in extracting structured information from unstructured

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 249–260, 2023.
https://doi.org/10.1007/978-3-031-44693-1_20



The Yorkshire Post

The Yorkshire Post is a daily broadsheet newspaper, published in Leeds in northern England. ... Alongside The Scotsman it is one of the flagship titles owned by Johnston Press ... Its focus on international and national news gives it a wider focus than that usually associated with a provincial newspaper; editions are available throughout the United Kingdom...

Fig. 1. A document in the DocRED dataset. New relations between entities are added by our revising. However, in the previous incomplete DocRED dataset, these triples are not in the ground truths.

textual data. Early studies focused primarily on the intra-sentence relation setting. Recently, there have been significant efforts on document-level RE, which deals with multiple mentions for each entity and complex inter-sentence relations. Consequently, the complexity of data annotation for document-level RE has increased dramatically. It is nearly impossible to rely completely on manual annotation to obtain large-scale, well-annotated datasets. To overcome this challenge, existing datasets (e.g., DocRED) employ a recommend-revise annotation scheme. This involves recommending relation triple candidates via distant supervision and then confirming their correctness through annotators. However, distant supervision relies on a sparse and incomplete knowledge base, which causes that human annotations fail to cover all ground-truth relation triples and suffer a major drawback - noisy labels[1]. For example, the relation between "Yorkshire Post" and "United Kingdom", i.e., "Country of origin", can be easily retrieved from the document in Fig. 1, whereas this triple is not included in DocRED. Current studies [5,8,13] have identified that the noisy labeled data is a key factor hindering the performance improvement of document-level RE. Nevertheless, how to resolve this issue has received limited attention.

Recent efforts in addressing noisy labeled data can be divided into two genres. The first and most direct solution is from the standpoint of data annotation. Huang et al. [8] pointed out the false negative problem[2] in DocRED, and assigned two expert annotators to relabel 96 documents from scratch. Subsequently, Tan et al. [13] adopted a human-in-the-loop approach to iteratively re-annotate 4,053 documents in DocRED. This approach involved reintroducing relation triples that were initially missed back into the original dataset. However, the complexity of the document-level RE task inevitably increases the difficulty and cost of producing high-quality benchmark datasets. Another economical solution is denoising from the model-based perspective. Reinforcement learning [1] and generative adversarial learning [5] successively solved the noise in the data. Compared with the data annotation solution, this model-based denoising method produces a smaller workload and higher reusability. Therefore, it is essential to study the problem of incompletely annotated data from a model perspective.

In this paper, we propose a novel **P**ositive-**G**uided **K**nowledge **D**istillation (**PGKD**) model for document-level RE. PGKD is based on a teacher-student architecture. The student model is partially supervised by the teacher model

[1] Noisy labels refer to incorrect or inaccurate annotations assigned to the samples.

[2] The relation triples are not in the ground truths of the dataset.

overlooking the NA (*no_relation*) instances. The proposed model works in a positive-guided manner. The distillation algorithm transfers the positive-class patterns to the student model. Although the student model is trained with all samples, it can avoid the pattern collapse [9] due to the supervision of the teacher model. Specifically, based on the soft label and prediction of the student model, we calculate a decoupled knowledge distillation loss. Moreover, the student model can benefit from a hard loss that enables it to learn the ground truths. Lastly, by incorporating both the hard loss and the knowledge distillation loss, the student model can learn from the teacher model's expertise while refining its own classification capabilities. In addition, we construct a new dataset called Mix-DocRED, consisting of both noisy labeled training data and well-labeled validation data. The evaluation of PGKD is performed on Mix-DocRED to validate our motivation. We can summarize our contributions as follows:

- We address the problem of the noisy labeled data in document-level RE from a model perspective, and design a new teacher-student architecture.
- We innovatively utilize positive samples to train the teacher model to ensure that the student model can avoid the pattern collapse and mimic the outputs of the teacher model on the positive classes.
- Experimental results on Mix-DocRED demostrate that PGKD achieves state-of-the-art performance for document-level relation extraction with noisy labeled data. Furthermore, PGKD outperforms existing competitive baselines even on the well-annotated Re-DocRED dataset.

2 Related Work

Document-Level Relation Extraction. There are two categories of approaches for document-level RE. On the one hand, researchers construct a delicately designed document graph [2]. Following this, many studies integrated similar structural dependencies to model documents. Otherwise, a special reasoning network was designed for relation inference [17]. On the other hand, there are some works [15,20] that attempt to use pre-trained language models directly for document-level RE without involving graph structure. Xu et al. [15] incorporated entity structure dependencies within the Transformer encoding part and throughout the overall system. Zhou et al. [20] introduced an adaptive-thresholding loss and a localized context pooling technique. These Transformer-based approaches are simple but very effective. However, most works were proposed under the assumption that datasets are completely annotated. Recently, Huang et al. [8] identified the false negative issue in DocRED and re-annotated 96 documents. Moreover, Tan et al. [13] adopted a more efficient semi-manual approach to re-annotate 4,053 documents. Despite their effectiveness, these annotation methods [8,13] are time-consuming and impractical for large-scale datasets. Therefore, we introduce the positive-guided knowledge distillation approach to address the problem of noisy labeled data in document-level RE.

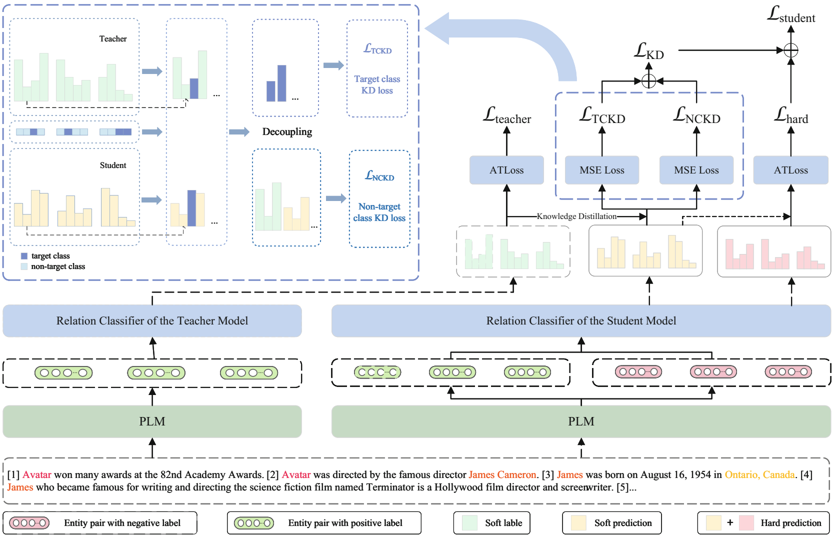

Teacher
Student
target class
non-target class
Decoupling
$\mathcal{L}_{\text{TCKD}}$
Target class KD loss
$\mathcal{L}_{\text{NCKD}}$
Non-target class KD loss
$\mathcal{L}_{\text{student}}$
$\mathcal{L}_{\text{KD}}$
$\mathcal{L}_{\text{teacher}}$
$\mathcal{L}_{\text{TCKD}}$
$\mathcal{L}_{\text{NCKD}}$
$\mathcal{L}_{\text{hard}}$
ATLoss
MSE Loss
MSE Loss
ATLoss
Knowledge Distillation
Relation Classifier of the Teacher Model
Relation Classifier of the Student Model
PLM
PLM
[1] Avatar won many awards at the 82nd Academy Awards. [2] Avatar was directed by the famous director James Cameron. [3] James was born on August 16, 1954 in Ontario, Canada. [4] James who became famous for writing and directing the science fiction film named Terminator is a Hollywood film director and screenwriter. [5]...
Entity pair with negative label
Entity pair with positive label
Soft lable
Soft prediction
+ Hard prediction

Fig. 2. The overall architecture of PGKD. The left part and the right part represent the teacher model and the student model, respectively. The part enclosed by the purple dotted box represents the process of obtaining the target and non-target knowledge distillation losses.

Knowledge Distillation. Hinton et al. [7] first introduced the concept of knowledge distillation. Its core idea is to transfer "dark knowledge" of the teacher model to the student model. Afterwards, Heo et al. [6] skipped redundant information that adversely affects the compression of the student model. Mirzadeh et al. [11] introduced a teacher assistant as an intermediary between teachers and students. Differing from the above methods, Zhao et al. [19] proposed the new concept of decoupled knowledge distillation (DKD) which comprises two components: target class knowledge distillation (TCKD) and non-target class knowledge distillation (NCKD). TCKD focuses on transferring knowledge related to the "difficulty" of training samples, whereas NCKD plays a crucial role in the effectiveness of logit distillation. Recently, Tan et al. [12] attempted to address the disparity between human-annotated data and distantly supervised data by knowledge distillation.

3 Problem Formulation

Given a set of n entities $\{e_1, \ldots, e_n\}$ in a document D, the goal of the document-level RE task is to predict all relation types $r \in R \cup \{no\_relation\}$ for each entity pair (e_h, e_t). e_h and e_t is the head and the tail entities, respectively. R stands for a collection of predefined relation classes. The setting of this work is to employ the incompletely labeled training set to train a document-level RE model and then evalutae this model with a well-annotated test set.

4 Model Architecture

As shown in Fig. 2, the proposed PGKD model is based on a teacher-student architecture. Specifically, we first follow Zhou et al. [20] to get the entity pair embedding with the local contextual representations. Next, we only apply positive samples to train the teacher model, which allows it to learn complete and clean positive-class patterns without interference from false negative samples. For the student model, we utilize all samples as the training data to generate the predicted results under the assistance of the teacher model and the guidance of the ground truths.

4.1 Teacher Model

Specifically, we first follow Zhou et al. [20] to obtain an entity pair representation that incorporates the localized context. The entity pair encoding method is equally applicable to the teacher model and the student model. Given a document $D = [h_l]_{l=1}^{L}$ containing L words, we utilize a special token "*" which is inserted before and after each mention. This approach allows us to easily identify the specific locations of mentions within the document. The document D is subsequently encoded using a pre-trained language model (PLM) to obtain its contextual embedding $H = [h_1, \ldots, h_L]$, $h_l \in \mathbb{R}^d$ of each token and cross token attention A. We adopts the vector representation of the marker "*" before the mention as the embedding of the mention m_j^i, where m_j^i represents the j^{th} mention of the i^{th} entity. All mentions to the same entity are adopted the logsumexp pooling to get the entity embedding e_i. The local contextual embedding $c_{h,t}$ of the entity pair is computed as follows:

$$c_{h,t} = H^T \frac{A_h \circ A_t}{A_h^T A_t}, \tag{1}$$

where A_h, A_t denote the attentions of the head and the tail entities, A_h, $A_t \in \mathbb{R}^L$, respectively. Then we compute the entity pair embedding $g^{(h,t)} \in \mathbb{R}^d$ as follows:

$$\begin{aligned} z_h &= \tanh\left(W_h e_h + W_{c_h} c_{h,t}\right), \\ z_t &= \tanh\left(W_t e_t + W_{c_t} c_{h,t}\right), \\ g_i^{(h,t)} &= \sum_{j=1}^{k} \left(z_h^j W_{g_i}^j z_t^j\right) + b_i, \\ g^{(h,t)} &= \left[g_1^{(h,t)}, g_2^{(h,t)}, \ldots, g_d^{(h,t)}\right] \end{aligned} \tag{2}$$

where $W_h, W_{c_h}, W_t, W_{c_t}, W_{g_i}^j$ and b_i are model parameters, $W_h, W_{c_h}, W_t, W_{c_t} \in \mathbb{R}^d, W_{g_i}^j \in \mathbb{R}^{d/k \times d/k}$. z_h and z_t indicate the embeddings of the head entity and the tail entity, $z_h, z_t \in \mathbb{R}^d$, respectively.

For the teacher model, the entity pair embedding $g_t^{(h,t)}$ of a positive sample is fed into a feed-forward linear layer to get the predicted result $P_t\left(r|e_h, e_t\right)$:

$$P_t\left(r|e_h, e_t\right) = \sigma\left(W_t g_t^{(h,t)} + b_t\right), \tag{3}$$

where W_t and b_t are the learnable parameters, $W_t, b_t \in \mathbb{R}^d$. σ is an activation function (e.g., *Sigmoid*). Because different entity pairs or classes have distinct interpretations for the same predicted score, a global threshold is inadequate. Therefore, we employ the adaptive-thresholding loss function [20] for the teacher model. A TH class is introduced to distinguish between positive and negative classes. Positive classes P_T are expected to have probabilities higher than TH, whereas negative classes N_T should have probabilities lower than TH. The loss function of the teacher model is as shown below:

$$\begin{aligned}\mathcal{L}_{teacher} = &- \sum_{r \in P_T} \log \left(\frac{\exp\left(logit_r\right)}{\sum_{r' \in P_T \bigcup \{TH\}} \exp\left(logit_{r'}\right)} \right) \\ &- \log \left(\frac{\exp\left(logit_{TH}\right)}{\sum_{r' \in N_T \bigcup \{TH\}} \exp\left(logit_{r'}\right)} \right),\end{aligned} \tag{4}$$

where *logit* is the hidden representation in the last layer before *Sigmoid*.

4.2 Student Model

The student model utilizes both positive and negative samples as training data. It generates a hard loss and a knowledge distillation loss, supervised by the true labels and the teacher model, respectively. Similarly to the teacher model, the student model first obtains the entity pair embedding $g_s^{(h,t)}$. Then the probability score $P_s\left(r|e_h, e_t\right)$ is obtained by inputting $g_s^{(h,t)}$ into a linear layer:

$$P_s\left(r|e_h, e_t\right) = \sigma\left(W_s g_s^{(h,t)} + b_s\right), \tag{5}$$

where W_s and b_s are the learnable parameters, $W_s, b_s \in \mathbb{R}^d$. Subsequently, we regard the ground-truth label of this input sample as the hard label and adopt the adaptive-thresholding loss function to compute the hard loss $\mathcal{L}_{hard}$, which can optimize its performance on individual sample classification.

Additionally, a knowledge distillation (KD) loss is introduced to obtain the supervised knowledge of the teacher on the student. We assign separate weights to the TCKD and NCKD losses. The true labels of the entity pair and the rest of the 97 relation types are considered as the target and the non-target classes, respectively. Specifically, we feed a positive sample to the trained teacher model that generates a soft label $P_t\left(r|e_h, e_t\right)$, while the student model produces a corresponding soft prediction $P_s\left(r|e_h, e_t\right)$. We then calculate the target class mask M_{TC} and non-target class mask M_{NC} based on the ground truth L_{PT}. The target class soft label L_{TC}, non-target class soft label L_{NC}, target class soft prediction P_{TC} and non-target class soft prediction P_{NC} can be available as follows:

$$\begin{aligned}L_{TC}\left(r|e_h, e_t\right) &= M_{TC} \cdot P_t\left(r|e_h, e_t\right), \\ L_{NC}\left(r|e_h, e_t\right) &= M_{NC} \cdot P_t\left(r|e_h, e_t\right), \\ P_{TC}\left(r|e_h, e_t\right) &= M_{TC} \cdot P_s\left(r|e_h, e_t\right), \\ P_{NC}\left(r|e_h, e_t\right) &= M_{NC} \cdot P_s\left(r|e_h, e_t\right)\end{aligned} \tag{6}$$

Table 1. Statistics of Re-DocRED and Mix-DocRED.

	Re-DocRED	Mix-DocRED		
data sets	Train	Train	Dev	Test
#Documents	3053	3053	500	500
Avg. #Entities	19.4	19.5	19.4	19.6
Avg. #Triples	28.1	12.5	34.6	34.9
Avg. #Sentences	7.99	7.9	8.2	7.9
#Positive samples	67,808	35,615	13,362	13,672
NA rate	93.5%		96.0%	

Ultimately, we employ the mean squared error loss function to calculate the target class KD loss $\mathcal{L}_{TCKD}$ and non-target class KD loss $\mathcal{L}_{NCKD}$. The student model can adequately learn the logit distribution of the teacher model on the positive classes. The positive-class pattern collapse of the student model can be avoided in this positive-guided manner.

$$\begin{aligned}\mathcal{L}_{TCKD} &= \frac{1}{|\mathcal{E}^t|}\sum_{(e_h,e_t)\in\mathcal{E}^t}\sum_{r\in\mathcal{R}}\left(L_{TC}\left(r|e_h,e_t\right)-P_{TC}\left(r|e_h,e_t\right)\right)^2,\\ \mathcal{L}_{NCKD} &= \frac{1}{|\mathcal{E}^t|}\sum_{(e_h,e_t)\in\mathcal{E}^t}\sum_{r\in\mathcal{R}}\left(L_{NC}\left(r|e_h,e_t\right)-P_{NC}\left(r|e_h,e_t\right)\right)^2,\end{aligned} \tag{7}$$

where $\mathcal{E}^t$ means the number of positive samples. $\mathcal{R}$ represents all predicted relation types. The KD loss is reformulated into a weighted sum of $\mathcal{L}_{TCKD}$ and $\mathcal{L}_{NCKD}$ as follows:

$$\mathcal{L}_{KD} = \alpha * \mathcal{L}_{TCKD} + \beta * \mathcal{L}_{NCKD}, \tag{8}$$

where the hyper-parameters α and β are utilized to disentangle the classical knowledge distillation process. By incorporating both the hard loss $\mathcal{L}_{hard}$ and the knowledge distillation loss $\mathcal{L}_{KD}$, the student model aims to learn from the teacher model's expertise while refining its own classification capabilities. Its final loss function is shown as follows:

$$\mathcal{L}_{student} = \gamma * \mathcal{L}_{hard} + \delta * \mathcal{L}_{KD}, \tag{9}$$

where γ and δ are the hyper-parameters to make trade-offs.

5 Experiments

5.1 Datasets

DocRED[3] [16] is a well-known benchmark dataset for document-level RE, but it is plagued by a high rate of false negative samples. To overcome this limitation,

[3] https://github.com/thunlp/DocRED.

Tan et al. [13] performed a re-annotation of the 4,053 documents in DocRED, creating a new dataset called Re-DocRED[4]. However, there is currently no reliable benchmark dataset for document-level denoising RE. So we fuse DocRED and Re-DocRED to construct a new dataset named Mix-DocRED, which comprises the training set of DocRED, as well as the dev and testing sets of Re-DocRED. Table 1 provides the statistics of Re-DocRED and Mix-DocRED.

5.2 Implementation Details

In this work, we employed BERT$_{Base}$ [3] and RoBERTa$_{Large}$ [10] as document encoders. AdamW was used as the optimizer of our model. We performed warm-up [4] on the initial 6% steps during training and set the learning rates to 5e-5 and 3e-5 for BERT$_{Base}$ and RoBERTa$_{Large}$, respectively. We performed the grid search on the development set to optimize the hyper-parameters, which include α, β, γ, and δ. The values for these hyper-parameters were 2, 1, 0.7, and 1, respectively. We reported the mean results with three different seeds. A single NVIDIA RTX 3090 GPU was used for all experiments. Precision, recall, Ign F1 and F1 scores served as the primary evaluation metrics.

5.3 Baselines

We conducted two sets of comparisons to evaluate the proposed PGKD model. Firstly, we compared PGKD against some existing competitive baselines on the DocRED leaderboard[5]. These baselines were developed under the assumption that the dataset is well-annotated, including BiLSTM [16], GAIN [17], ATLOP [20], and DocuNet [18]. Secondly, we compared PGKD with SSR-PU [14], the current state-of-the-art framework for document-level denoising RE. SSR-PU was a unified positive-unlabeled learning framework that effectively solved the incomplete labeling problem.

5.4 Main Results

We report the mean and standard deviation of PGKD on the Mix-DocRED test set compared to other strong baselines. As seen in Table 2, PGKD outperforms the competitive models, achieving the highest F1 score of 56.50 and 59.92, respectively. To facilitate a more direct comparison with the latest denoising framework SSR-PU, we adopt ATLOP as the backbone. Our PGKD outperforms SSR-PU by 0.36 and 0.42 F1 points, respectively. This implies that PGKD is superior to SSR-PU in the RE ability from incomplete annotated data. Furthermore, It is worth noting that BiLSTM, GAIN, ATLOP, and DocuNet experience significant drops in F1 score when facing with incomplete labeling scenarios. For instance, DocuNet shows a decrease of 10.51 and 10.18 F1 points compared to PGKD. The conspicuous gap between these baselines and PGKD is mainly due

[4] https://github.com/tonytan48/Re-DocRED.

[5] https://competitions.codalab.org/competitions/20717.

Table 2. Experimental results (%) on the Mix-DocRED test set. Results with † are reported from [14]. Results with $*$ are based on our implementations.

Model	Ign F1	F1	Precision	Recall
BiLSTM†	32.57 ± 0.22	32.86 ± 0.22	77.04 ± 1.01	20.89 ± 0.17
GAIN+BERT$^{\dagger}_{Base}$ [17]	45.57 ± 1.36	45.82 ± 1.38	88.11 ± 1.07	30.98 ± 1.36
ATLOP+BERT$^{*}_{Base}$ [12]	45.18 ± 0.23	45.48 ± 0.25	85.66 ± 0.30	30.96 ± 0.28
DocuNet+BERT$^{\dagger}_{Base}$ [18]	45.88 ± 0.33	45.99 ± 0.33	**94.16 ± 0.32**	30.42 ± 0.29
SSR-PU+ATLOP+BERT$^{\dagger}_{Base}$ [14]	55.21 ± 0.12	56.14 ± 0.12	70.42 ± 0.18	46.67 ± 0.14
PGKD+BERT$^{*}_{Base}$	**55.45 ± 0.20**	**56.50 ± 0.21**	65.85 ± 0.21	**49.50 ± 0.22**
GAIN+RoBERTa$^{*}_{Large}$ [17]	48.65 ± 0.24	48.76 ± 0.25	88.60 ± 0.25	33.64 ± 0.26
ATLOP+RoBERTa$^{*}_{Large}$ [12]	48.70 ± 0.30	48.91 ± 0.30	89.68 ± 0.32	33.63 ± 0.35
DocuNet+RoBERTa$^{*}_{Large}$ [18]	49.54 ± 0.27	49.74 ± 0.25	**94.81 ± 0.26**	34.27 ± 0.27
SSR-PU+ATLOP+RoBERTa$^{\dagger}_{Large}$ [14]	58.68 ± 0.43	59.50 ± 0.45	74.21 ± 0.53	49.67 ± 0.77
PGKD+RoBERTa$^{*}_{Large}$	**58.87 ± 0.24**	**59.92 ± 0.25**	67.61 ± 0.25	**53.79 ± 0.23**

Table 3. Error distributions of ATLOP and PGKD on the dev set of Mix-DocRED. In each cell, the data on the left (or right)

Predictions	Ground Truth				
		$r \in R$		NR	
	$r \in R$	C 3,523 (25.49%)	6,203 (35.61%)	MR 457 (3.31%)	4,055 (23.28%)
		W 2,021 (14.67%)	2,928 (16.81%)		
	NR	MS 7,818 (56.57%)	4,231 (24.29%)	CN 179,413	175,815

to that the former prioritize precision over recall, at the cost of sacrificing overall performance. Without the ability to systematically identify relation triples that are overlooked in the dataset, these baselines simply treat unlabeled data as negative samples. Fortunately, our PGKD is able to learn clean positive-class patterns that aid in better distinguishing between positive and negative samples. PGKD overcomes the challenge posed by noisy labeled data and appropriately increases the recall score, leading to an overall improvement in performance.

Additionally, while the decreased precision of PGKD is indeed a concern, it can be attributed to that the teacher model monitors the student model. Since the teacher model is trained only on positive samples, the student model tends to be biased towards predicting more positive samples under its guidance. Nevertheless, despite the decrease in precision, our model (PGKD) exhibits significant improvement over the baselines according to recall and F1 scores. Thus, PGKD outperforms the competitive baselines by effectively balancing the trade-off between precision and recall scores.

Table 4. Results (%) on the revised Re-DocRED test set. Results with † are reported from [13]. Results with $*$ are based on our implementations.

Model	Ign F1	F1
ATLOP+BERT$^*_{Base}$ [12]	73.14	73.86
DocuNet+BERT$^*_{Base}$ [18]	73.94	74.03
PGKD+BERT$^*_{Base}$	**74.28**	**74.35**
ATLOP+RoBERTa$^\dagger_{Large}$ [12]	76.94	77.73
DocuNet+RoBERTa$^\dagger_{Large}$ [18]	77.27	77.92
KD-DocRE+RoBERTa$^\dagger_{Large}$ [12]	77.63	78.35
PGKD+RoBERTa$^*_{Large}$	**77.67**	**78.38**

5.5 Error Analysis

In this section, we follow Tan et al. [12] and provide a detailed error analysis to specify ATLOP and PGKD. We compare the predictions with the ground truths to form five categories, including: 1) **C**: where all predicted relations are correct. 2) **W**: the entity pair is correctly identified, but there are certain predicted relations that are incorrect. 3) **MS**: where the model fails to identify the entity pair in the ground truth. 4) **MR**: where the model generates a relation label for a negative sample. 5) **CN**: where both the head entity and tail entity are not included in the ground truth and the predicted relation does not correspond to any existing relation. Afterwards, we design a confusion matrix in Table 3 to present each predicted category's number and score. Given that the final evaluation score is assessed based on $r \in R$ triples, the **CN** category is ignored when calculating the final score. Based on the above matrix, we can draw conclusions. Table 3 indicates that the sum of error scores under the **MR** and **MS** categories for ATLOP is 59.88%, exceeds PGKD's sum by 12.31%. This comparison proves that PGKD outperforms ATLOP when determining the relation between head and tail entities. Moreover, PGKD's score under the **C** category is 35.61% (6,203), while ATLOP's score is 25.49% (3,523), further indicating that PGKD has a higher recall score than ATLOP. These strong contrasts demonstrate the effectiveness of positive-guided knowledge distillation in addressing the problem of noisy labeled data.

5.6 Experiment on the Well-Annotated Re-DocRED

In this section, we assess the performance of PGKD in comparison with various state-of-the-art baselines, namely ATLOP, DocuNet, and KD-DocRE [12], utilizing the well-annotated Re-DocRED for both training and testing. As shown in Table 4, PGKD achieves a superior F1 score of 78.38%, outperforming the other baselines. Despite being designed to handle noisy labels, PGKD shows relative improvement over the baselines even in well-annotated scenarios. Additionally, these results can be regarded as a maximum value for document-level RE with incompletely annotated data.

Table 5. Experimental results (%) of positive relation classification on the Mix-DocRED and Re-DocRED test sets. Results with $*$ are based on our implementation.

Model	Mix-DocRED		Re-DocRED	
	Ign F1	F1	Ign F1	F1
Teacher+BERT$^*_{Base}$	78.45	78.56	90.97	91.36
Teacher+RoBERTa$^*_{Large}$	79.85	80.19	92.62	92.95

5.7 Analysis and Discussion on Positive Relation Classification

We conduct an experiment to explore the eligibility of a model trained solely with positive samples to be the teacher model. The experiment focuses on classifying relation types for positive samples, referred to as positive relation classification (PRC). The number of positive samples for each set on Mix-DocRED and Re-DocRED is summarized in Table 1. The results in Tables 2 and 5 show that the teacher model on Mix-DocRED exceeds PGKD by 22.06 (78.56 vs. 56.50) and 20.27 (80.19 vs. 59.92) F1 scores, respectively. This strongly proves that the teacher model has effectively learned patterns of positive samples, making it a valuable source of knowledge for the student model. The PRC metric, which excludes all NA samples, provides an upper bound of the RE performance for a given dataset. The performance gap between PRC and RE should not be significant if the dataset has good annotation quality. However, we observe that the performance on the standard RE task is inferior to counterpartrpar on the PRC task, suggesting that the annotations for positive samples are of higher quality compared to the entire dataset. Therefore, it is reasonable to use a model trained only with positive samples as the teacher model.

6 Conclusion

In this work, we propose a novel PGKD model to address the noisy labeled data problem for document-level RE. Our model is based on the teacher-student architecture. PGKD innovatively only utilizes positive samples to train the teacher model. The student model is partially supervised by the teacher model to avoid positive class pattern collapse and interference of noisy labeled data. We conducted experiments on two distinct datasets, Mix-DocRED and Re-DocRED. Extensive experimental results demostrate that the proposed PGKD exhibits SOTA effectiveness in denoising noisy labeled data, outperforming competitive baselines.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 62276095 and 72204261), the National Social Science Foundation of China (Grant No. 20&ZD047), and the Hunan Provincial Natural Science Foundation of China (Grant No. 2021JJ40681).

References

1. Chen, J., Fu, T., Lee, C., Ma, W.: H-FND: hierarchical false-negative denoising for distant supervision relation extraction. In: ACL/IJCNLP, pp. 2579–2593 (2021)
2. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In: EMNLP-IJCNLP, pp. 4924–4935 (2019)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
4. Goyal, P., et al.: Accurate, large minibatch SGD: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
5. Hao, K., Yu, B., Hu, W.: Knowing false negatives: an adversarial training method for distantly supervised relation extraction. In: EMNLP, pp. 9661–9672 (2021)
6. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: IEEE/CVF, pp. 1921–1930 (2019)
7. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
8. Huang, Q., Hao, S., Ye, Y., Zhu, S., Feng, Y., Zhao, D.: Does recommend-revise produce reliable annotations? an analysis on missing instances in docred. In: ACL, pp. 6241–6252 (2022)
9. Li, T., Hu, Y., Ju, A., Hu, Z.: Adversarial active learning for named entity recognition in cybersecurity. Comput. Mater. Continua **66**(1) (2021)
10. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
11. Mirzadeh, S., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: AAAI, pp. 5191–5198 (2020)
12. Tan, Q., He, R., Bing, L., Ng, H.T.: Document-level relation extraction with adaptive focal loss and knowledge distillation. In: Findings of ACL, pp. 1672–1681 (2022)
13. Tan, Q., Xu, L., Bing, L., Ng, H.T.: Revisiting docred - addressing the overlooked false negative problem in relation extraction. arXiv preprint arXiv:2205.12696 (2022)
14. Wang, Y., Liu, X., Hu, W., Zhang, T.: A unified positive-unlabeled learning framework for document-level relation extraction with different levels of labeling. In: EMNLP, pp. 4123–4135 (2022)
15. Xu, B., Wang, Q., Lyu, Y., Zhu, Y., Mao, Z.: Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In: IAAI, EAAI, pp. 14149–14157 (2021)
16. Yao, Y., et al.: Docred: a large-scale document-level relation extraction dataset. In: ACL, pp. 764–777 (2019)
17. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: EMNLP, pp. 1630–1640 (2020)
18. Zhang, N., et al.: Document-level relation extraction as semantic segmentation. In: IJCAI, pp. 3999–4006 (2021)
19. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: IEEE/CVF, pp. 11943–11952 (2022)
20. Zhou, W., Huang, K., Ma, T., Huang, J.: Document-level relation extraction with adaptive thresholding and localized context pooling. In: AAAI, pp. 14612–14620 (2021)

PromptCL: Improving Event Representation via Prompt Template and Contrastive Learning

Yubo Feng, Lishuang Li(✉), Yi Xiang, and Xueyang Qin

Dalian University of Technology, Dalian, Liaoning, China
lils@dlut.edu.cn

Abstract. The representation of events in text plays a significant role in various NLP tasks. Recent research demonstrates that contrastive learning has the ability to improve event comprehension capabilities of Pre-trained Language Models (PLMs) and enhance the performance of event representation learning. However, the efficacy of event representation learning based on contrastive learning and PLMs is limited by the short length of event texts. The length of event texts differs significantly from the text length used in the pre-training of PLMs. As a result, there is inconsistency in the distribution of text length between pre-training and event representation learning, which may undermine the learning process of event representation based on PLMs. In this study, we present PromptCL, a novel framework for event representation learning that effectively elicits the capabilities of PLMs to comprehensively capture the semantics of short event texts. **PromptCL** utilizes a **Prompt** template borrowed from prompt learning to expand the input text during **C**ontrastive **L**earning. This helps in enhancing the event representation learning by providing a structured outline of the event components. Moreover, we propose Subject-Predicate-Object (SPO) word order and Event-oriented Masked Language Modeling (EventMLM) to train PLMs to understand the relationships between event components. Our experimental results demonstrate that PromptCL outperforms state-of-the-art baselines on event related tasks. Additionally, we conduct a thorough analysis and demonstrate that using a prompt results in improved generalization capabilities for event representations (Our code will be available at https://github.com/YuboFeng2023/PromptCL).

Keywords: Event representation · Prompt learning · Contrastive learning

1 Introduction

Distributed event representations are a widely-used machine-readable representation of events, known to capture meaningful features relevant to various applications [1,2,13]. Due to event texts are too short, capturing their semantic relationships is a challenging task. For example, despite the greater lexical overlap

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 261–272, 2023.
https://doi.org/10.1007/978-3-031-44693-1_21

between "military launch program" and "military launch missile" their semantic similarity is limited, but "military launch program" and "army starts initiative" display a certain degree of semantic similarity, even though they share no lexical overlap.

In previous studies [3–5,19], Neural Tensor Networks (NTNs) [16] have been commonly utilized to construct event representations by composing the constituent elements of an event, i.e., (subject, predicate, object). Nevertheless, these approaches entail a substantial compositional inductive bias and are inadequate for handling events that possess additional arguments [6]. Recent research has demonstrated the effectiveness of employing powerful PLMs, such as BERT [9], to create flexible event representations instead of using static word vector compositions [17,21]. Nevertheless, utilizing PLMs alone to learn event representations is insufficient to capture the complicated relationships between events. Therefore, in order to tackle the challenge of capturing complex relations between events, some researchers have proposed the use of graph neural networks in event representation learning. This approach has shown to yield better performance, as demonstrated in recent studies [22,23]. But graph neural networks are often associated with high computational complexity, which can lead to significant difficulties in training the models [24]. To fully capture complicated event relations and efficiently learn event representations, Gao et al. proposed SWCC [6], which leverages contrastive learning [7] to improve the event comprehension ability of PLMs. It has achieved state-of-the-art results on event similarity tasks and downstream tasks.

In our work, we argue that there is a rich amount of event comprehension ability in PLMs, but previous works did not make fully use of such abilities. Inspired by the advancements in prompt learning [14,15], we have realized that providing task descriptions to PLMs can help to elicit the knowledge embedded within them. And then, this knowledge can be utilized to enhance event representation learning. To learn event representations, previous works [6,21] leverage contrastive learning to improve the event comprehension capacity of PLMs. However, they share three common limitations. Firstly, the length of event text is relatively short, which differs significantly from the text length used in the pre-training of language models. As a result, the distribution of text length between pre-training and event representation learning is inconsistent. This inconsistency may undermine the learning process of event representation based on PLMs. Secondly, the Predicate-Subject-Object (PSO) word order, which is adopted by PLMs-based event representation models [6,21], is significantly different from the natural language word order used during the pre-training [9]. In PSO word order, the inversion of subject and predicate can potentially undermine the performance, as the pre-trained MLM knowledge may have a counterproductive effect. Because MLM in pre-training predicts a masked token based on its context [9], a change in word order can also cause the position of the context to change. Therefore, the pre-trained MLM knowledge may be a burden for event representation learning in PSO word order. Finally, the state-of-the-art model utilizes MLM loss to prevent the forgetting of token-level knowledge during the training of event representation in the PLM [6]. However, the model only randomly masks one sub-word, which may not provide sufficient understanding of complex event texts [20].

We are motivated to address the above issues with the goal of eliciting the event comprehension capabilities from PLMs. To this end, we present **PromptCL**: a **Prompt** template-based **C**ontrastive **L**earning framework for event representation learning. To address the first issue, we propose a novel prompt template-based contrastive learning method. In this approach, we incorporate a prompt template borrowed from prompt learning into contrastive learning, which comprises a description outlining the event components. The injection of the prompt template serves two purposes: it extends the length of event texts and provides semantic guidance to PLMs. To address the second issue, we propose using the SPO word order to solve the problem of subject-predicate inversion, which aligns with the natural language word order. To address the final issue, we present an approach called EventMLM, which focuses on the structure of events and aims to increase the masking rate. EventMLM not only masks entire words but also masks the complete subject, predicate, or object of the event. This approach trains PLMs to understand the relationships between event components. Overall, our study makes the following noteworthy contributions:

- We propose **PromptCL**, a simple and effective framework that improves event representation learning using PLMs. To the best of our knowledge, this is the first study that utilizes prompt learning and contrastive learning to elicit event representation abilities from PLMs.
- We introduce prompt template-based contrastive learning that extends the length of event texts and provides semantic guidance to PLMs. Additionally, we introduce the SPO word order and the EventMLM method, which are designed to train PLMs to comprehend the relationships between event components.
- Our experimental results demonstrate that our framework outperforms previous state-of-the-art methods on event-related tasks. We conduct a thorough analysis of the proposed methods and demonstrate that they generate similarity scores that are more closely aligned with the ground truth labels.

2 The Proposed Approach

This section details our proposed approach that aims to enhance event representations by eliciting the event comprehension capabilities of PLMs. Our approach is illustrated in Fig. 1, comprising three parts: the prompt template-based contrastive learning (left), and the SPO word order (middle), and the EventMLM (right).

2.1 Prompt Template-Based Contrastive Learning

The proposed contrastive learning method based on prompt templates involves augmenting an event text by randomly inserting a template using a Bernoulli

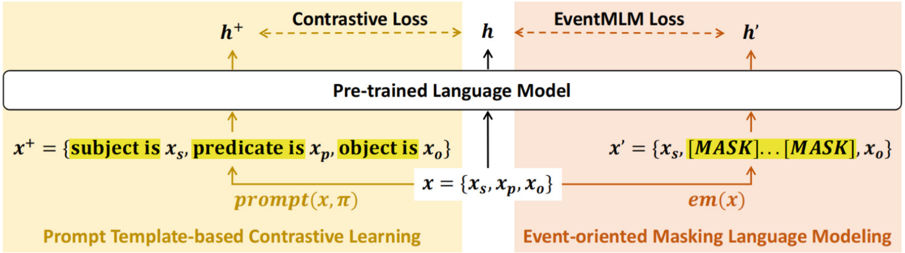

Contrastive Loss
EventMLM Loss
h^+
h
h'
Pre-trained Language Model
$x^+ = \{$subject is x_s, predicate is x_p, object is $x_o\}$
$x' = \{x_s, [MASK] \ldots [MASK], x_o\}$
$x = \{x_s, x_p, x_o\}$
$prompt(x, \pi)$
$em(x)$
Prompt Template-based Contrastive Learning
Event-oriented Masking Language Modeling

Fig. 1. Architecture of PromptCL.

distribution. This augmentation results in a modified event that serves as a positive example during the training of contrastive learning.

Prompt Template. Given an event $x = \{x_s, x_p, x_o\}$, the function $prompt(\cdot)$ inserts a template into the event with a probability π following a Bernoulli distribution:

$$x^+ = prompt(x, \pi) \tag{1}$$

and the resulting prompt-augmented event is denoted as x^+:

$$x^+ = \{subject\ is\ \underline{x_s}, predicate\ is\ \underline{x_p}, object\ is\ \underline{x_o}\} \tag{2}$$

For example, if $x = \{military, launch, missile\}$, the augmented event x^+ could be :

$$x^+ = \{subject\ is\ \underline{military}, predicate\ is\ \underline{launch}, object\ is\ \underline{missile}\} \tag{3}$$

The random insertion of the template ensures that the model is trained on a slightly diverse set of events, improving its ability to capture the core semantic meaning of events.

Dual Positive Contrastive Learning. To augment the impact of prompt templates and enhance the diversity of positive examples, we introduce an additional positive sample, whereby an input event x_i is compared not only with its prompt-augmented text $x^+_{i,1} = prompt(x_i, \pi)$, but also with another prompt-augmented text $x^+_{i,2} = prompt(x_i, \pi)$. Drawing inspiration from [10] and based on in-batch negatives [7], we extend InfoNCE objective [12] to:

$$\mathcal{L}_{CL} = \sum_{\alpha=\{h^+_{i,1}, h^+_{i,2}\}} -\log \frac{g(h_i, \alpha)}{g(h_i, \alpha) + \sum_{k \in \mathcal{N}(i)} g(h_i, h_k)} \tag{4}$$

where $h^+_{i,1}$ and $h^+_{i,2}$ correspond to event representations of $x^+_{i,1}$ and $x^+_{i,2}$ respectively. $k \in \mathcal{N}(i)$ is the index of in-batch negatives and $g(\cdot)$ is a function: $g(h_i, h_k) = \exp(h_i^\top h_k / \tau)$, where $\tau \in R^+$ is temperature.

2.2 Subject-Predicate-Object Word Order

Unlike prior studies [6,21], where PSO word orders were used to construct the input events, we use the Subject-Predicate-Object (SPO) word order in our study. The event text x consists of three components, namely the subject x_s, predicate x_p, and object x_o. Specifically, we utilize the PLM to process an event text that consists of a sequence of tokens, following the input format represented below:

$$[\text{CLS}]\ x_s\ x_p\ x_o\ [\text{SEP}] \tag{5}$$

Let $s = [s_0, s_1, ..., s_L]$ be an input sequence, where s_0 corresponds to the [CLS] token and s_L corresponds to the [SEP] token. When given an event text as input, a PLM generates a sequence of contextualized vectors:

$$[v_{[\text{CLS}]}, v_{x_1}, ..., v_{x_L}] = \text{PTM}(x) \tag{6}$$

The representation of the [CLS] token, denoted by $v_{[\text{CLS}]}$, serves as the first input to downstream tasks in many PLMs. Typically, the final representation of an input sequence is obtained by taking the [CLS] representation as the output, that is, $h = v_{[\text{CLS}]}$.

2.3 Event-Oriented Masking Language Modeling

To fully utilize the text comprehension ability of PLMs, we present a novel event-oriented masking function, denoted by $em(\cdot)$, which randomly masks a component of the input event using a uniform distribution. For a given event $x = \{x_s, x_p, x_o\}$, the resulting masked event is denoted as x':

$$x' = em(x) \tag{7}$$

For example, if the predicate x_p is randomly selected to be masked using a uniform distribution, we replace it with special tokens [MASK]. Note that multiple tokens may be replaced by the [MASK] tokens.

$$x' = \{x_s, [\text{MASK}]...[\text{MASK}], x_o\} \tag{8}$$

Distinctively, our EventMLM method differs from previous work [6], which merely masks a single token. Our proposed method not only masks several tokens but also considers the components of the event. Moreover, our method focuses on the event structure and trains the PLM to comprehend the relationships between the components, thus enhancing the event representation. In this example, the PLM needs to accurately predict the masked tokens (predicate x_p) by understanding the semantic relationship between the subject x_s and object x_o.

2.4 Model Training

The overall training objective comprises three terms:

$$\mathcal{L}_{overall} = \mathcal{L}_{CL} + \mathcal{L}_{EventMLM} + \mathcal{L}_{CP} \tag{9}$$

Firstly, we have the prompt template-based contrastive learning loss ($\mathcal{L}_{CL}$), which effectively incorporates prompt templates into event representation learning. Secondly, the EventMLM loss ($\mathcal{L}_{EventMLM}$) aims to improve the text comprehension ability of PLMs and teaches the model to comprehend the relationships between the components of input events. Finally, we introduce the prototype-based clustering objective ($\mathcal{L}_{CP}$) as an auxiliary loss to cluster the events while enforcing consistency between cluster assignments produced for different augmented representations of the input event [6].

3 Experiments

Consistent with conventional practices in event representation learning [3,6,19,21], we conduct an analysis of the event representations acquired through our approach on two event similarity tasks and one transfer task.

3.1 Dataset and Implementation Details

For the event representation learning models training and event similarity tasks, we utilize the datasets released by Gao et al. [6][1]. For the transfer task, we use the MCNC dataset that was previously employed by Lee and Goldwasser [11][2]. It is noteworthy that above datasets explicitly specify the components of the event, indicating that they support the arbitrary organization of word order.

Our model begins with the checkpoint of BERT-based-uncased [9], and we utilize the [CLS] token representation as the event representation. During training, we employe an Adam optimizer with a batch size of 256. The learning rate for the event representation model is set to 2e-7. The value of temperature is set to $\tau = 0.3$. Furthermore, we select the probability of prompt template insertion to be $\pi = 0.2$.

3.2 Event Similarity Tasks

Hard Similarity Task. The objective of the hard similarity task is to assess the ability of the event representation model to differentiate between similar and dissimilar events. Weber et al. [19] created a dataset (referred to as"Original") comprising two types of event pairs: one with events that have low lexical overlap but should be similar, and the other with events that have high overlap but should be dissimilar. The dataset consists of 230 event pairs. Ding et al. [3] subsequently expanded this dataset to 1,000 event pairs (denoted as "Extended"). We evaluate the performance of our model on this task using Accuracy(%) as the metric, which measures the percentage of instances where the model assigns a higher cosine similarity score to the similar event pair compared to the dissimilar one.

[1] https://github.com/gaojun4ever/SWCC4Event.

[2] https://github.com/doug919/multi_relational_script_learning.

Table 1. Evaluation performance on the similarity tasks. The Hard Similarity Task is represented by the Original and Extended datasets. The Transitive Sentence Similarity is evaluated using the Transitive dataset.

Model	Original(%)	Extended(%)	Transitive(ρ)
Event-comp [19]	33.9	18.7	0.57
Predicate Tensor [19]	41.0	25.6	0.63
Role-factor Tensor [19]	43.5	20.7	0.64
NTN-IntSent [3]	77.4	62.8	0.74
UniFA [21]	75.8	61.2	0.71
UniFA-S [21]	78.3	64.1	0.75
HeterEvent$_{[W+E]}$ [22]	76.6	62.3	0.73
MulCL$_{[W+E]}$ [23]	78.3	64.3	0.76
SWCC [6]	80.9	72.1	0.82
PromptCL(Ours)	**81.7**	**78.7**	**0.82**

Transitive Sentence Similarity. This dataset [8] (denoted as ”Transitive”) is comprised of 108 pairs of transitive sentences containing a singular subject, object, and verb (e.g., “military launch missile”). Each pair is annotated with a similarity score ranging from 1 to 7, with higher scores indicating greater similarity between the two events. To evaluate the performance of the models, we employ Spearman's correlation(ρ) to measure the relationship between the predicted cosine similarity and the manually annotated similarity score, consistent with prior work in the field [3,19,21].

Comparison Methods. In our study, we conduct a comparative analysis of our proposed approach with various baseline methods. We group these methods into four distinct categories:

(1) Neural Tensor Networks: The models, **Event-comp** [19], **Role-factor Tensor** [19], **Predicate Tensor** [19], and **NTNIntSent** [3], employ Neural Tensor Networks to learn event representations. (2) Pre-trained Language Model: Two event representation learning frameworks that leverage PLMs are **UniFAS** [21] and **UniFA-S** [21]. (3) Graph Neural Network: The utilization of graph neural networks for event representation learning is employed by **HeterEvent** [22] and **MulCL** [23]. (4) Contrastive Learning: **SWCC** [6] is a state-of-the-art framework that is based on a PLM and combines contrastive learning and clustering.

Results. Table 1 presents the results of various methods on the challenging similarity tasks, including hard similarity and transitive sentence similarity. The findings reveal that the proposed PromptCL outperforms other approaches in terms of performance. Compared to the UniFA-S approach that simply utilizes PLMs, PromptCL exhibits superior performance due to its innovative features

Table 2. Evaluation performance on the MCNC task. *: results reported in Gao, et al. [6]

Model	Accuracy(%)
Random	20.00
PPMI*	30.52
BiGram*	29.67
Word2vec*	37.39
SWCC [6]	44.50
PromptCL(Ours)	**47.06**

such as prompt template and contrastive learning that better explore the text comprehension ability. PromptCL outperforms state-of-the-art event representation methods, such as SWCC, that leverage a PLM and contrastive learning. The observed enhancements can be attributed to PromptCL's thorough exploration of PLM's text comprehension capabilities via its prompt template-based contrastive learning and EventMLM techniques. This finding emphasizes the limited exploration of text comprehension ability in prior research and underscores the efficacy of our proposed framework, PromptCL.

3.3 Transfer Task

We conduct an evaluation of the generalization ability of event representations on the Multiple Choice Narrative Cloze (MCNC) task, which involves selecting the next event from a small set of randomly drawn events, given a sequence of events. We adopt the zero-shot transfer setting to ensure comparability with prior research [6].

Results. The performance of various methods on the MCNC task is reported in Table 2. The table indicates that the PromptCL method exhibits the highest accuracy on the MCNC task in the unsupervised setting. This result suggests that PromptCL has superior generalizability to downstream tasks compared to other methods in the study. We believe that the use of a prompt template can enhance the generalization capabilities of event representation models, as discussed in the section "Content of prompt".

4 Analysis

Ablation Study. To evaluate the effectiveness of each component in the proposed approach, we conduct an ablation study as presented in Table 3. We begin by investigating the impact of the prompt template method by setting the probability of inserting templates to zero. Removing the prompt template component

Table 3. Ablation study for several methods evaluated on the similarity tasks. *: degenerate to PSO word order.

Model	Original(%)	Extended(%)	Transitive(ρ)
PromptCL	81.7	78.7	0.82
w/o Prompt Template	80.0(−1.7)	70.8(−7.9)	0.81(−0.01)
w/o SPO Word Order*	79.9(−1.8)	74.1(−4.6)	0.80(−0.02)
w/o EventMLM	80.9(−0.8)	76.7(−2.0)	0.76(−0.06)
BERT(InfoNCE)	72.1	63.4	0.75

resulted in a significant drop of 7.9 points in the model's performance on the extended hard similarity task. Furthermore, we examine the effect of the SPO word order method on the similarity tasks. Removing this component led to a drop of 1.8 points in the model's performance on the original hard similarity task. We also study the impact of the EventMLM method. Removing this component causes a 0.06 (maximum) point drop in performance on the transitive sentence similarity task. The BERT (InfoNCE) is trained using the InfoNCE objective only.

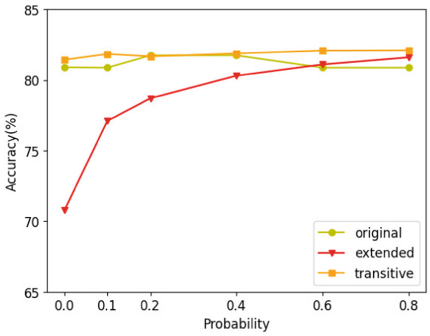


Fig. 2. Effect of prompt insertion probability.

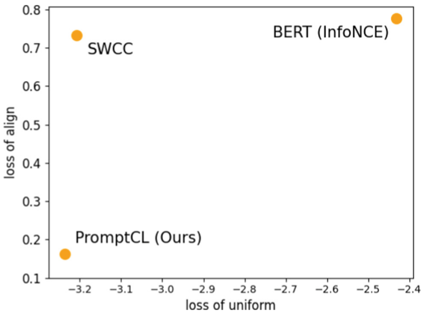


Fig. 3. To plot the align loss and uniform loss. (lower is better)

Probability of Prompt Insertion. The influence of the probability of inserting prompt templates during the training process is depicted in Fig. 2. We have observed that as the probability of prompt template insertions increases, the model's overall performance in intrinsic evaluation steadily improves. The insertion of prompt templates during the training of event representation models enhances the generalization of event representation in intrinsic evaluation tasks.

Table 4. To demonstrate the semantic clarity of prompts and evaluate their performance.

prompt	Original(%)	Extended(%)	Transitive(ρ)
x_s x_p x_o	80.0	70.8	0.81
subject x_s predicate x_p object x_o	79.1	72.4	0.82
subject : x_s predicate : x_p object : x_o	**81.7**	76.7	0.82
subject is x_s predicate is x_p object is x_o	**81.7**	**78.7**	**0.82**

Uniformity and Alignment. Figure 3 displays the uniformity and alignment of various event representation models, along with their Transitive Sentence Similarity results. In general, models that exhibit better alignment and uniformity achieve superior performance, which confirms the findings in Wang, et al. [18]. Additionally, we observe that the insertion of prompt templates during event representation learning significantly improves alignment compared to baselines.

Content of Prompt. Table 4 illustrates the impact of adjusting prompt content on the training process. As shown in the table, an increase in prompt semantic clarity results in a better performance on the Hard Similarity Tasks. The generalization of event representation models is closely related to the clarity of the prompt template used during training. Specifically, a clearer prompt template provides enhanced semantic guidance, leading to more effective event representation models with better generalization capabilities.

Table 5. A case study on the Extended dataset of Hard Similarity Task.

Event A	Event B	BERT (InfoNCE)	PromptCL (Ours)	Label
we focus on issues	he pay attention to problems	0.46	0.61	1
we focus on issues	we focus on people	0.72	0.57	0
he flee city	i leave town	0.62	0.72	1
he flee city	he flee hotel	0.68	0.41	0
he explain things	she analyze problems	0.44	0.50	1
he explain things	he explain love	0.51	0.31	0

Case Study. Table 5 shows the case study of randomly sampling several groups of events from the Extended dataset of the Hard Similarity Task. The performance of BERT(InfoNCE) and PromptCL in predicting the similarity scores of these events was evaluated. A closer alignment between the predicted and ground truth similarity scores indicates a deeper understanding of the event by the model. The results are presented in Table 5, which demonstrate that PromptCL outperforms BERT(InfoNCE) in predicting similarity scores that more closely align with the ground truth labels. This suggests that the proposed prompt template-based contrastive learning, and SPO word order, and EventMLM can aid in comprehending short event texts and provide semantic guidance for PLMs.

5 Conclusion

This study presents a novel framework called PromptCL, which aims to improve the learning of event representations through the use of PLMs, without the need for additional features such as co-occurrence information of events as used in SWCC. In particular, we introduce a prompt template-based contrastive learning method and SPO word order that allow us to easily elicit the text comprehension ability of PLMs, and an EventMLM method that trains the PLM to comprehend the relationships between event components. Our experiments demonstrate that PromptCL achieves superior performance compared to state-of-the-art baselines on several event-related tasks. Moreover, our comprehensive analysis reveals that the utilization of a prompt leads to enhanced generalization capabilities for event representations.

Acknowledgments. This work is supported by grants from the National Natural Science Foundation of China (No. 62076048), the Science and Technology Innovation Foundation of Dalian (2020JJ26GX035).

References

1. Chen, H., Shu, R., Takamura, H., Nakayama, H.: GraphPlan: story generation by planning with event graph. In: Proceedings of the 14th International Conference on Natural Language Generation (2021)
2. Deng, S., et al.: OntoED: low-resource event detection with ontology embedding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (2021)
3. Ding, X., Liao, K., Liu, T., Li, Z., Duan, J.: Event representation learning enhanced with external commonsense knowledge. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
4. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
5. Ding, X., Zhang, Y., Liu, T., Duan, J.: Knowledge-driven event embedding for stock prediction. In: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (2016)
6. Gao, J., Wang, W., Yu, C., Zhao, H., Ng, W., Xu, R.: Improving event representation via simultaneous weakly supervised contrastive learning and clustering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2022)
7. Gao, T., Yao, X., Chen, D.: SimCSE: Simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021)

8. Kartsaklis, D., Sadrzadeh, M.: A study of entanglement in a categorical framework of natural language. In: Proceedings of the 11th workshop on Quantum Physics and Logic (2014)
9. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT (2019)
10. Khosla, P., et al.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems (2020)
11. Lee, I.T., Goldwasser, D.: Multi-relational script learning for discourse relations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
12. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
13. Rezaee, M., Ferraro, F.: Event representation with sequential, semi-supervised discrete variables. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2021)
14. Schick, T., Schütze, H.: Exploiting cloze-questions for few-shot text classification and natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (2021)
15. Schick, T., Schütze, H.: It's not just size that matters: small language models are also few-shot learners. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2021)
16. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
17. Vijayaraghavan, P., Roy, D.: Lifelong knowledge-enriched social event representation learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (2021)
18. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning (2020)
19. Weber, N., Balasubramanian, N., Chambers, N.: Event representations with tensor-based compositions. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
20. Wettig, A., Gao, T., Zhong, Z., Chen, D.: Should you mask 15% in masked language modeling? In: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (2023)
21. Zheng, J., Cai, F., Chen, H.: Incorporating scenario knowledge into a unified fine-tuning architecture for event representation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
22. Zheng, J., Cai, F., Ling, Y., Chen, H.: Heterogeneous graph neural networks to predict what happen next. In: Proceedings of the 28th International Conference on Computational Linguistics (2020)
23. Zheng, J., Cai, F., Liu, J., Ling, Y., Chen, H.: Multistructure contrastive learning for pretraining event representation. IEEE Trans. Neural Networks Learn. Syst. (2022)
24. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI open (2020)

RMGCN: Masked Graph Convolutional Networks for Relation-Aware Entity Alignment with Dangling Cases

Xinyu Liu[1,2](✉), Feng Zhou[1,2], and X. Y. Li[3]

[1] Key Laboratory of Intelligent Telecommunications Software and Multimedia,
Beijing, China
{liuxinyu2017,zfeng}@bupt.edu.cn

[2] School of Computer Science, Beijing University of Posts and Telecommunications,
Beijing, China

[3] School of Cyberspace Security, Beijing University of Posts and
Telecommunications, Beijing, China
lixiaoyong@bupt.edu.cn

Abstract. Entity alignment (EA) is to match entities referring to identical real-world facts among different knowledge graphs (KGs). For simplicity, most previous work ignores the existence of dangling entities in the source KG. However, entity alignment task with dangling entities has emerged as a novel demand in real scenarios. Some work explores new dangling-aware loss functions and transfers existing methods on dangling settings, whose performance is still not satisfactory. Thus, in this work, we propose Relation-aware Masked Graph Convolutional Networks (RMGCN). In the learning stage, it can not only take advantage of the masking mechanism to alleviate the negative impact from nearby dangling neighbors, but also take both graph structure and relations of KGs into consideration. In the inference stage, it performs a two-step alignment which firstly filters out dangling entities, and then align the remaining entities. We adopt a novel distribution-based dangling entity detection method in first step to decrease the error propagation from dangling detection to the following EA task. The experimental results show that our model outperforms all state-of-the-art models with at least 7.5% on F1 score.

Keywords: Knowledge graph · Entity alignment · Dangling entity

1 Introduction

Knowledge graphs (KGs) have been widely applied in NLP tasks [4]. Various KGs have been created, yet each was constructed independently, resulting in a considerable amount of complementary and redundant information across them. Thus, efforts are being made to integrate overlapping KGs via the equivalent entities among them to provide a more comprehensive knowledge for downstream applications. One common way for doing this is through entity alignment (EA), which

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 273–285, 2023.
https://doi.org/10.1007/978-3-031-44693-1_22

aims to identify equivalent entities from different KGs. In recent years, work on embedding-based entity alignment methods [3,13,17,18] is growing rapidly. This kind of approach takes entity embeddings as training data, and makes matchable entities closer in a low-dimension vector space.

Previous entity alignment methods almost assume that every entity in source KG has a counterpart in target KG [3,14,19]. However, in real scenarios, the target KG is unlikely to contain all entities in the source KG, which poses a challenge on previous work in terms of practicality. To address this issue, Sun et al. [12] define such entities as dangling entities and re-propose a more practical entity alignment task. Specifically, the method needs to detect whether a given entity is dangling before conducting entity alignment.

Most recently, researchers have explored entity alignment in new settings, but there is still significant room for improvement. One issue is that some studies [10] rely on the textual descriptions of KGs rather than the basic graph features, e.g., graph structures and relations. Many KGs do not contain textual information, highly depending on which will undermine the generalizability of models. Another issue is that high-performing methods that solely leverage the basic graph information are mostly based on the classic entity alignment model MTransE [3], which cannot model complex relations in KGs. Though GNN-based methods outperform MTransE in conventional entity alignment task, they cannot be easily transferred to such a new task due to the ability to learn more dangling neighbor information. In other words, whether GNN-based methods are suitable under new settings have not been well tapped yet. Additionally, in inference stage, previous work [12] adopts a simple but non-robust estimation method to detect dangling entities.

Considering the aforementioned issues, we propose a framework with dangling entity settings called **R**elation-aware **M**asked **G**raph **C**onvolutional **N**etwork (RMGCN). It mainly consists of four modules, i.e., a structure-aware entity embedding module, a relation-aware entity embedding module, a loss function module, and an inference module. Specifically, in the first two modules, we focus on utilizing GNNs to model complex information of KGs, i.e., graph structures and relations between entities, respectively. In the loss function module, we adopt dangling detection loss, alignment loss and NCA loss to optimize the embedding. Finally, in the inference module, we perform a two-step inference to identify dangling entities and conduct entity alignment. To be more specific, the main contributions are summarized as follows:

- We propose a masked GCNs which can capture rich graph features, and eliminate negative influence of dangling entities. A simple and effective relation information aggregation method is also proposed to obtain the relation-aware entity embeddings.
- We propose a novel distribution-based estimation method to detect dangling entities in the inference module, which is robust to the data distribution.
- We evaluate our approach on DBP2.0, where the experimental results demonstrate the improvement of RMGCN, comparing to state-of-the-art methods. The code is available online[1].

[1] https://github.com/RMGCN/RMGCN-code

The rest of this paper is organized as follows. Section 2 discusses some related work. Section 3 states the problem formulation. Section 4 introduces our approach in detail. Section 5 describes experimental settings and Sect. 6 reports and analyzes experimental results. Finally, we summarizes our contributions in Sect. 7.

2 Related Work

Embedding-Based Entity Alignment. The main idea of embedding-based approaches is to encode KGs into a low-dimensional vector space and then obtain the entity embedding. Based on the way of constructing entity embeddings, EA models can be classified into translation-based model and GNN-based model.

Translation-based embedding methods [1,6,16] represent a relation between entities as a translation from the source entity to the target entity. Among them, TranE [1] is prevalent and widely used in EA models [3,13,15,21]. However, TranE-based entity alignment model has limits in capturing complex relations, which pushes the adoption of GNN-based methods that can utilize the information about the neighbors of entities. GCN-Align [17] is the first GNN-based model that embeds KGs into a unified vector space via GCNs. AGEA [22] equally weights relations of entities to integrate relation information. AliNet [14] considers both the neighbor and far-neighbor information by attention mechanism. HMAN [20] uses BERT to capture extra description information, such as attribute values and entity names. However, such semantic information is not common to all KGs, leading to poor generalizability of the model.

Overall, GNN-based EA models are explosively growing these years due to their powerful ability to capture graph structures. They generally outperform translation-based methods in conventional entity alignment tasks.

Dangling Entity Detection. Sun et al. [12] first introduce and define the dangling entity problem, then construct DBP2.0 dataset that include dangling entities. They novelly propose three dangling entity detection loss functions. By minimizing those loss functions, the distance between dangling entities and their nearest neighbors will increase. Liu et al. [9] utilize not only the nearest neighbors of the source entity but also those of the target entity to help the detection, and regard the entity alignment as the optimal transportation (OT) problem. Luo et al. [10] also convert EA task into an OT problem and introduce additional semantic information. However, as previously mentioned, such information reduces the generalization ability of models.

GNN-based methods demonstrate superior performance in conventional EA tasks but aggregate more negative information from dangling entities in dangling-aware tasks. As a result, nearly all existing approaches avoid GNN-based methods. Although some methods [9,12] claim that their models are interchangeable, the GNN-based models are less effective than MTransE. Therefore, this paper focuses on GNNs and pays more attention to leveraging the basic graph information that is common among all KGs.

3 Problem Formulation

Formally, a KG is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$ denote a set of entities, relations and triples, respectively. The set of triples is defined as $\mathcal{T} = \{(e_h, r, e_t) | e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$, where the relation r directs from e_h to e_t. Different from the conventional entity alignment task that requires no dangling entities in $\mathcal{E}$, we regard $\mathcal{E}$ as $\mathcal{D} \cup \mathcal{M}$, where $\mathcal{D}$ and $\mathcal{M}$ are composed of dangling entities and matchable ones, respectively. Assume there are two KGs, representing the source KG and the target one, denoted as $\mathcal{G}_s = (\mathcal{E}_s, \mathcal{R}_s, \mathcal{T}_s)$ and $\mathcal{G}_t = (\mathcal{E}_t, \mathcal{R}_t, \mathcal{T}_t)$, respectively. Given a set of dangling entities $\mathcal{D}_s \subset \mathcal{E}_s$, and a set of equivalent entities between $\mathcal{E}_s$ and $\mathcal{E}_t$, i.e., $\mathcal{L} = \{(e_i, e_j) \in \mathcal{E}_s \times \mathcal{E}_t \ \mid e_i \equiv e_j\}$, our goal is to determine whether a given entity $e_s \in \mathcal{E}_s$ is dangling. If it is not, we will find its counterpart $e_t \in \mathcal{E}_t$.

4 Our Approach: RMGCN

4.1 Overview

As we mentioned in Sect. 1 and Sect. 2, previous work either paid more attention on textual features of given KGs that results in poor generalization ability, or paid less attention on adopting GNN-based methods that can handle dangling entities. This leaves a gap for us by adopting GNN-based methods to handle entity alignment with dangling cases while considering purely graph structures and relations. To this end, we propose a GNN-based method, named *Relation-aware Masked Graph Convolutional Networks* (RMGCN). Figure 1 illustrates its overall architecture. As we can see, based on the given source KG and target KG, it firstly initializes a random unified entity embeddings *X_init*. To take advantage of features hidden in graph structures and relations, we have designed and implemented two vital processes to obtain structure-aware entity embedding and relation-aware entity embedding based on *X_init*. Then, the two generated embeddings will be concatenated as *Z*. And we minimize three loss functions alternately to optimize it. Finally, based on *Z*, our methods can perform a two stage inference, i.e., a novel distribution-based dangling entity detection and an entity alignment task. We will detail our method in the following.

4.2 Structure-Aware Entity Embedding

The graph structure which is represented by connectivity between entities is the fundamental information of KGs, which plays a vital role in entity alignment task. Graph Convolutional Networks (GCNs) [2] can learn entity embeddings with such type of information, whose effectiveness is proven by several previous work [17,18]. Inspired by them, we utilize an L-layer modified GCNs to encode entities in each KGs into a unified low-dimensional vector space. Specifically, the input of the l-th layer is a set of entity embeddings $X^l = \{x^l_{e_1}, x^l_{e_2}, ..., x^l_{e_n}\}$, where n is the total number of entities of two KGs and $x^l_{e_i}$ is the embedding of

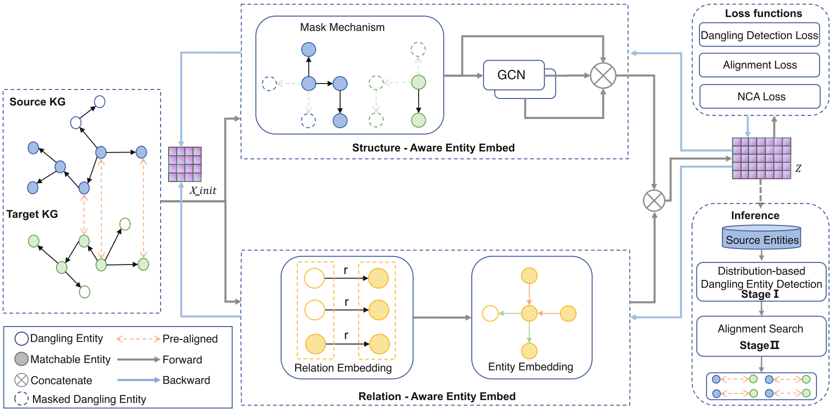


Fig. 1. Overview architecture of RMGCN.

entity e_i. The input X^0 of the first layer is X_init. The output of the l-th layer is:

$$X^{l+1} = \xi(\sigma(\hat{A}X^l W^l + b)), \tag{1}$$

where ξ is an L2 normalization, σ is an activation function chosen as `tanh()`, $\hat{A} = A + I$, A is a modified adjacency matrix that we called masked adjacency matrix, and I is an identity matrix, W^l is the trainable weight matrix of X^l, and b is a bias vector.

Considering that dangling entities and matchable ones share the same weight matrix in GCNs, which may negatively impact the encoding of matchable entities. So we need to force GCNs to learn as little information as possible about the dangling neighbors of an entity. Thus, we propose a mask mechanism which can mask dangling entities in the input entity set to construct masked adjacency matrix A. The elements of A is defined as follows:

$$a_{ij} = \begin{cases} 1, & (e_i, r, e_j) \in \mathcal{T} \text{ and } e_i, e_j \notin \mathcal{D} \text{ or } (e_j, r, e_i) \in \mathcal{T} \text{ and } e_i, e_j \notin \mathcal{D} \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where a_{ij} refers to the element at the i-th row and j-th column, $\mathcal{T}$ and $\mathcal{D}$ refer to the set of triples and dangling entities, respectively, as we introduced in Sect. 3. In this way, the connections between dangling entities and others will be masked.

It is worth to note that the above mask mechanism will not lead to information loss for dangling entities, because we re-introduce such information in dangling detection loss function to optimize the embeddings later (see Sect. 4.4). Additionally, we remove the diagonal node degree matrix in primal GCNs and calculate vanilla adjacency matrix with masked triples instead of constructing an adjacency matrix which contains complex relation information like [17]. This is because we will utilize degree information and aggregate the relation information with a non-shared weights in other module (see Sect. 4.3).

Finally, to better obtain the global structures of KGs, we concatenate the initial embedding and the output embeddings of each layer of GCNs as the final output of masked GCNs, named as the structure-aware entity embedding:

$$X = X^0 \parallel X^1 \parallel X^2 \parallel \ldots \parallel X^L. \tag{3}$$

4.3 Relation-Aware Entity Embedding

As mentioned in Sect. 4.2, masked GCNs learn graph structure information of KGs via adjacency matrix. However, they did not take the relation information into consideration. Inspired by previous work [18,22,23], we propose a simple but efficient relation information aggregation approach parallel to the masked GCNs. Specifically, we define a relation embedding V as: $V = \{v_{r_1}, v_{r_2}, ..., v_{r_m}\}$, which is composed of v_{r_i} that is calculated as the average of its head and tail entity embeddings:

$$v_{r_i} = \frac{\sum_{(e_h,\ r,\ e_t) \in \mathcal{T}_{r_i}} (x_{e_h} + x_{e_t})}{2 \cdot |\mathcal{T}_{r_i}|}, \tag{4}$$

where $\mathcal{T}_{r_i}$ is a subset of $\mathcal{T}$, whose relation in each elements meets $r = r_i$.

Intuitively, the surrounding relations of two matchable entities tend to be similar, thus we introduced the relation-aware entity embedding $Y = \{y_{e_1}, y_{e_2}, ..., y_{e_n}\}$. Before diving into it, we should notice that relations are directional for an entity, and the direction plays a vital role for embedding. Moreover, those rare relations are more important to an entity. For example, comparing two entities with a single and multiple outgoing relations, respectively, the outgoing relation of the former entity should be paid more attention. Thus, we have designed a weighted average aggregation method. Different from previous work [22] that do not distinguish the weight of outgoing and ingoing relations when calculating the average, for an entity x_{e_i}, we calculate its relation-aware embedding separately according to outgoing and ingoing relations:

$$y_{e_i} = \sum_{(e_h,\ r,\ e_t) \in \mathcal{T}_h} \frac{v_r}{|\mathcal{T}_h|} + \sum_{(e_h,\ r,\ e_t) \in \mathcal{T}_t} \frac{v_r}{|\mathcal{T}_t|}, \tag{5}$$

where $\mathcal{T}_h = \{(e_h, r, e_t) \in \mathcal{T} | e_h = e_i\}$ and $\mathcal{T}_t = \{(e_h, r, e_t) \in \mathcal{T} | e_t = e_i\}$.

Through Sect. 4.2 and Sect. 4.3, we obtained the structure-aware entity embeddings X and the relation-aware entity embeddings Y. The final step is to concatenate both of them:

$$Z = \{z_{e_1}, z_{e_2}, ..., z_{e_n}\} = \xi(X) \parallel \xi(Y), \tag{6}$$

where ξ is the normalization function and chosen as L2 normalization.

4.4 Loss Functions

As the right side of Fig. 1 illustrates, we optimize our model via three loss functions, i.e., dangling entity detection loss, entity alignment loss and NCA loss.

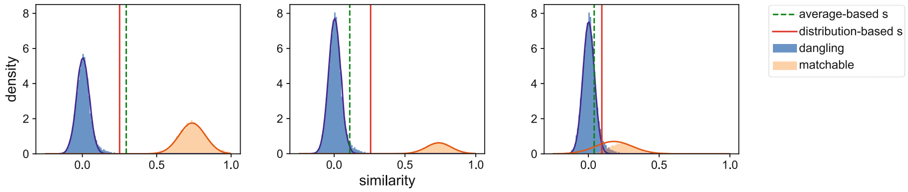
Specifically, marginal ranking loss [12] is chosen as the dangling entity detection loss which tries to keep dangling entities at least λ_1 away from their neighbors. And contrastive alignment loss [14] is chosen as the entity alignment loss that tends to minimize the distance between positive sample pairs and let the distance between negative sample pairs at least λ_2. And NCA loss [8] is adopted to alleviate the hubness problem [11].

4.5 Inference

As we detailed in Sect. 3, there are two steps in the inference stage. At first, for the input entities, our model should conduct the dangling entity detection to filter the dangling entities out. Then, for those remaining ones, our model will try to align them to the target KG.

Step I: Dangling Detection. We adopt a simple but effective cosine similarity between an entity and its nearest neighbor (NN) to determine whether it is dangling. For an entity, it will be labeled as matchable when the similarity scores between itself and its NN is greater than a threshold s. In this paper, we propose a distribution-based dangling entity detection method.

In previous work [9,12], s is calculated as the average of the similarity scores of all pairs of entities of their corresponding NN. Though this strategy can generate an acceptable performance, the results are still sub-optimal in some cases because the average score relies on the relative ratio between dangling entities and matchable ones. For example, if the source KG has a quite amount of dangling entities, the average similarity score will be closer to the average of the dangling entities, resulting in misclassification for some dangling entities.



(a) (b) (c)

Fig. 2. Toy examples for density plot and fitted distribution of NN similarity in different distribution of dangling entities and matchable ones. (a) The number of dangling entities is similar to matchable ones. (b) The number of dangling entities is approximately four times of matchable ones. (c) The distributions of dangling and matchable entities intersect.

To address this issue, we propose a novel method to determine s, which can eliminate the impact of data distribution to some extent. Intuitively, the probability density function of the above-mentioned similarity list will exhibit a central tendency. Figure 2 shows an example of NN similarity density, which has two obvious spikes and can be fitted by the sum of two normal distributions.

Table 1. Details of the datasets.

Datasets		# Entities	# Danglings	# Relations	# Triples	# Alignment
ZH - EN	Chinese	84,996	51,813	3,706	286,067	33,183
	English	118,996	85,813	3,402	586,868	
JA - EN	Japanese	100,860	61,090	3,243	347,204	39,770
	English	139,304	99,534	3,396	668,341	
FR - EN	French	221,327	97,375	2,841	802,678	123,952
	English	278,411	154,459	4,598	1,287,231	

Our method fit such overall distribution into two distributions of dangling entities and matchable entities. And the threshold s is chosen as the value of the x-axis of the intersection point of two distributions. Specifically, we consider normal distribution, Cauchy distribution and Laplace distribution to find the best match. Figure 2a is the density of NN similarity of training set. It shows that average-based and distribution-based s have similar performance in this situation. However in Fig. 2b, if the ratio of dangling entities is much more higher, the average-based s will generate some incorrect results. This deficiency is even more pronounced in the test set, because the distribution of matchable and danglings are closer. Figure 2c is an example to illustrate this phenomenon. Obviously, our approach is robust to imbalanced data distributions and exhibits improved performance as the model is trained more effectively.

Step II: Alignment. After removing dangling entities, our model will conduct entities alignment between two KGs. We still use cosine similarity to measure the distance between two entities. For an entity in source KG, our model adopts greedy search strategy to find an entity in the target KG with the highest cosine similarity as the alignment result.

5 Experimental Settings

We develop our model on TensorFlow framework, and all experiments are conducted on a Linux server with an Intel(R) Xeon(R) Platinum 8260 CPU @ 2.30 GHz, a NVIDIA RTX A6000 GPU and 86 GB memory.

Datasets. We evaluate RMGCN on DBP2.0 [12], which is built upon DBpedia [7] and contains three cross-lingual pairs of KGs: ZH-EN, JA-EN, and FR-EN. For each pair, it consists of triples across the two KGs, aligned entity links and dangling entities. The statistical details are listed in Table 1. Note that the datasets do not provide the name of entities and relations in real world which fundamentally eliminates name bias problem. On each dataset, we use 30% of the data for training, 20% for validation and 50% for testing.

Evaluation Metrics. We use relaxed evaluation and consolidated evaluation [12] to measure our proposed method. Specifically, relaxed evaluation only applies alignment strategy on matchable entities, which is similar to conventional entity alignment task. We use mean reciprocal rank (MRR) and Hits@K (where K is 1 or 10) to evaluate the performance. Consolidated evaluation is a two step evaluation which involves all entities. For each source entity, it will conduct a dangling entity detection first. If an entity is wrongly labeled, the incorrect prediction will be propagated to the following alignment result. We use precision, recall, and F1 score as the measurement metrics.

Compared Methods. As far as we know, there are only two work [9,12] proposed for DBP2.0, which are both focus on the dangling entity detection loss functions without improving embedding model. They all use MTransE [3] and Alinet [14] as the base embedding model. In this paper, we incorporate Marginal Ranking(MR) dangling detection loss proposed by Sun et al. [12], so we choose MTransE with MR and Alinet with MR as baselines for fairness.

Implementation Details. We initial the entity embedding using Xavier initialization [5] and choose 0.2 as dangling distance margin λ_1 and 1.4 as negative alignment distance margin λ_2 for all compared methods which use those. As for RMGCN, we use a 2-Layer masked GCNs in structure-aware entity embedding model. The input dimensions of GCNs are 512 and 384, respectively, and the output dimension is 256. The input and output dimension of relation-aware entity embedding module are 512. For training, we generate 30 negative samples for each entity and set the learning rate 0.0005. The batch size is 10240 and we use early stop to control the training epoch.

Model Variants. To evaluate the performance of components, we provide three implementation variants of our model: 1) w/o MGCN (our approach without mask mechanism); 2) w/o Relation-aware (our approach without relation-aware module); 3) w/o distribution-based detection (our approach with previous simple strategy [12] but without distribution-based dangling entity detection).

6 Results

In this section, we will illustrate the results on the previously mentioned relaxed evaluation and consolidated evaluation, as well as the ablation study.

6.1 Relaxed Evaluation

Table 2 shows the relaxed evaluation results on DBP2.0. As we can see, using the same Marginal Ranking (MR) loss, RMGCN achieves better scores on all metrics of all datasets. Such results indicate that our proposed method can prioritize the correct matches and hit the equivalent entity better, which can benefit the two-step final entity alignment performance. Such preliminary results demonstrate the effectiveness of our model in capturing basic graph information.

Table 2. Entity alignment results (relaxed evaluation) on DBP2.0.

Methods w/MR	ZH - EN			JA - EN			FR - EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
AliNet	0.343	0.606	0.433	0.349	0.608	0.438	0.230	0.477	0.312
MTransE	0.378	0.693	0.487	0.373	0.686	0.476	0.259	0.541	0.348
RMGCN	**0.465**	**0.718**	**0.555**	**0.468**	**0.711**	**0.554**	**0.316**	**0.571**	**0.402**

6.2 Consolidated Evaluation

Dangling Entity Detection. Table 3 shows the dangling entity detection results on DBP2.0. According to the results, our proposed framework outperforms other baselines both in recall and F1 scores but has a mediocre result in precision. This indicates that our approach is able to detect more dangling entities with fewer missed reports. Therefore, it can feed more real matchable entities to the entity alignment step. For the same reason, our method misclassifies those matchable entities that are vaguely identified as dangling entities, which leads to a slight decrease in precision. However, the higher F1 score shows that our method can detect dangling entities better than baselines overall.

Table 3. Dangling entity detection results on DBP2.0.

Methods w/MR	ZH - EN			JA - EN			FR - EN		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
AliNet	0.752	0.538	0.627	0.779	0.580	0.665	**0.552**	0.570	0.561
MTransE	**0.781**	0.702	0.740	**0.799**	0.708	0.751	0.482	0.575	0.524
RMGCN	0.737	**0.833**	**0.782**	0.743	**0.785**	**0.763**	0.514	**0.742**	**0.607**

Table 4. Two-step entity alignment results on DBP2.0.

Methods w/MR	ZH - EN			JA - EN			FR - EN		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
AliNet	0.207	0.299	0.245	0.231	0.321	0.269	0.195	0.190	0.193
MTransE	0.302	0.349	0.324	0.313	0.367	0.338	0.260	0.220	0.238
RMGCN	**0.476**	**0.379**	**0.422**	**0.438**	**0.401**	**0.419**	**0.396**	**0.259**	**0.313**

Two-step Entity Alignment. Table 4 shows the two-step entity alignment results. RMGCN obtains better performance in all metrics. Compared with the best baseline MTransE with MR, our method improves at least 12.5% on precision, 3% on recall and 7.5% on F1 score. The good performance benefits from both our efficient entity embedding model and our novel method to detect dangling entities. This proves that our entity embedding model can learn both structure and relation information efficiently and comprehensively, which makes the alignment more accurate. Moreover, the distribution-based dangling detection method leads to less error propagated to the following alignment stage.

6.3 Ablation Studies

To verify each module of our proposed method, we conduct three ablation studies on ZH-EN dataset of DBP2.0 (see the last part of Sect. 5). As shown in Table 5, compared to w/o MGCN, the improvement trend of RMGCN on both relaxed evaluation and consolidation evaluation is the same as the trend we reported above, which indicates the effectiveness of masked GCNs. Comparing w/o Relation-aware with RMGCN, we can see that almost every metrics have an obviously drop. This demonstrates that the relation information we aggregate do benefit the entity alignment. While it is normal that the drop of recall leads to a slight improve of precision on dangling detection. For distribution-based dangling entity detection, our method only affects consolidated evaluation, so it has the same level performance on relaxed evaluation, and we can see that RMGCN has a better performance on both dangling entity detection and entity alignment on consolidated evaluation.

Table 5. Ablation studies on ZH-EN dataset of DBP2.0.

Methods w/ MR	relaxed evaluation			consolidated evaluation (two-step entity alignment)			dangling detection		
	Hits@1	Hits@10	MRR	Precision	Recall	F1	Precision	Recall	F1
RMGCN	0.465	0.718	0.555	0.476	0.379	0.422	0.737	0.833	0.782
w/o MGCN	0.455	0.711	0.545	0.393	0.397	0.395	0.757	0.752	0.755
w/o Relation-aware	0.374	0.649	0.468	0.363	0.323	0.341	0.745	0.797	0.770
w/o distribution-based detection	0.468	0.718	0.556	0.440	0.401	0.419	0.751	0.794	0.772

7 Conclusions

In this paper, we propose a novel Relation-aware Masked Graph Convolutional Network for entity alignment task with dangling cases. By introducing mask mechanism to GCNs, our method embeds KGs with graph structures while eliminate the negative influence of dangling entities. To obtain the considerable relation information of KGs, we propose a relation aggregating approach to enhance entity embedding. In inference stage, we propose a novel distribution-based dangling entity detection method which is robust to data distributions. Comprehensive experiments show that our approach significantly outperforms state-of-art methods. Moreover, ablation studies verify the effectiveness of key components.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
2. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and deep locally connected networks on graphs. In: 2nd International Conference on Learning Representations, ICLR 2014
3. Chen, M., Tian, Y., Yang, M., Zaniolo, C.: Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. arXiv preprint arXiv:1611.03954 (2016)
4. Chen, M., Zhou, T., Zhou, P., Zaniolo, C.: Multi-graph affinity embeddings for multilingual knowledge graphs. In: AKBC@ NIPS (2017)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
6. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 687–696 (2015)
7. Lehmann, J., et al.: DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**, 167–195 (2015)
8. Liu, F., Chen, M., Roth, D., Collier, N.: Visual pivoting for (unsupervised) entity alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4257–4266 (2021)
9. Liu, J., et al.: Dangling-aware entity alignment with mixed high-order proximities. In: NAACL, pp. 1172–1184 (2022)
10. Luo, S., Yu, S.: An accurate unsupervised method for joint entity alignment and dangling entity detection. In: ACL, pp. 2330–2339 (2022)
11. Radovanovic, M., Nanopoulos, A., Ivanovic, M.: Hubs in space: popular nearest neighbors in high-dimensional data. J. Mach. Learn. Res. **11**, 2487–2531 (2010)
12. Sun, Z., Chen, M., Hu, W.: Knowing the no-match: entity alignment with dangling cases. In: ACL-IJCNLP, pp. 3582–3593 (2021)
13. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: Proceedings of 16th International Semantic Web Conference, pp. 628–644 (2017)
14. Sun, Z., et al.: Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 222–229 (2020)
15. Trisedya, B.D., Qi, J., Zhang, R.: Entity alignment between knowledge graphs using attribute embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 297–304 (2019)
16. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
17. Wang, Z., Lv, Q., Lan, X., Zhang, Y.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 349–357 (2018)
18. Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., Zhao, D.: Relation-aware entity alignment for heterogeneous knowledge graphs. arXiv preprint arXiv:1908.08210 (2019)

19. Xu, K., et al.: Cross-lingual knowledge graph alignment via graph matching neural network. In: Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL) (2019)
20. Yang, H.W., Zou, Y., Shi, P., Lu, W., Lin, J., Sun, X.: Aligning cross-lingual entities with multi-aspect information. In: EMNLP-IJCNLP, pp. 4431–4441 (2019)
21. Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y.: Multi-view knowledge graph embedding for entity alignment. arXiv preprint arXiv:1906.02390 (2019)
22. Zhu, R., Luo, X., Ma, M., Wang, P.: Adaptive graph convolutional network for knowledge graph entity alignment. In: EMNLP 2022, pp. 6011–6021 (2022)
23. Zhu, R., Ma, M., Wang, P.: RAGA: relation-aware graph attention networks for global entity alignment. In: Proceedings of Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, pp. 501–513 (2021)

SymCoNLL: A Symmetry-Based Approach for Document Coreference Resolution

Ying Mao[1,2](✉), Xinran Xie[1,2], Lishun Wang[1,2], Zheyu Shi[1,2],
and Yong Zhong[1,2]

[1] Chengdu Institute of Computer Application, Chinese Academy of Sciences,
Chengdu 610041, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
{maoying19,xiexinran19,wanglishun17,shizheyu21}@mails.ucas.ac.cn
zhongyong@casit.com.cn

Abstract. Coreference resolution aims to identify the referential relationships between all entities in a text. However, traditional rule-based or hand-crafted feature-based methods encounter difficulties with long sentences and complex structures. Although many neural end-to-end coreference resolution models have been proposed, there are still challenges in resolving coreferences among multiple sentences due to weak contextual connections and ambiguity caused by distant mentions. To address these problems, we propose a symmetry-based method called SymCoNLL that studies coreference resolution from two aspects: mention identification and coreference prediction. Specifically, at the local level, our method focuses on mention identification based on semantic, syntactic, and mention type features, and improves coreference prediction by enhancing the similarity between mentions using semantic and mention type features. At the global level, it strengthens the internal connections of mentions in the same cluster using symmetry. We validate the effectiveness of our method in mention identification and coreference prediction through experiments on OntoNotes 5.0 dataset. The results show that our method significantly outperforms the baseline methods.

Keywords: Mention Identification · Coreference Prediction · Coreference Resolution · Symmetry

1 Introduction

Coreference resolution [1–3] refers to the task of identifying referring expressions in a text that relate to the same entity. It is an important part of natural language processing (NLP), particularly in knowledge extraction and entity disambiguation. The task aims to identify all words or phrases which represent the same entity or thing in a text, and then merge them into one cluster. The longer the text, the more difficult the task.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 286–297, 2023.
https://doi.org/10.1007/978-3-031-44693-1_23

For document-level coreference resolution tasks, we need to consider the relationships between all noun phrases in the entire document, which faces several challenges: 1) contextual information loses its relevance with increasing text distance, resulting in weak connections between mentions in different sentences. 2) A noun phrase may be associated with multiple entities, and there may exist various complex relationships between these entities, requiring consideration of more contextual information. 3) It is necessary to resolve multiple possible coreference relationships within the same document. 4) The current standard pre-training model BERT only supports 512 tokens, making it difficult to handle document-level tasks. Our work focuses on document-level coreference resolution tasks and aims to address these challenges.

End-to-end coreference resolution methods simplify the coreference resolution process by reducing manual intervention and coordination between tasks, and are widely used in document-level coreference resolution tasks. End-to-end models [4] divide coreference resolution into two parts: mention identification and coreference prediction. In the mention identification phase, the model needs to identify all mentions in the text, including nouns, pronouns, etc., and find possible reference relationships among them. In the coreference prediction phase, the model needs to cluster the identified mentions to form several mention clusters, each cluster represents the same entity.

However, existing methods have not thoroughly analyzed the key factors that affect document-level coreference resolution, resulting in poor performance. In mention representation, [4–7] use pruning algorithms to cut off many correct mention candidates in exchange for sufficient memory space, which sacrifices the model's ability to fully explore mention features. In coreference representation, [4,5,8] only use semantics to calculate the possibility of clustering spans, leading to insufficient model features, while [6,9,10] increase the complexity of the model with non-critical features. When calculating coreference scores, [5,7,10,11] assume a unidirectional relationship between mention candidates and antecedents and believe that antecedents should appear before mention candidates in the text. This unidirectional and sequential assumption destroys the natural symmetry of coreference resolution tasks. In this paper, we posit that mentions within the same coreference cluster should hold an equal status, and the directionality of their relationships should not be determined by the order of their appearance. Furthermore, we suggest that entities inherently refer to each other. Finally, when making coreference predictions, [11,12] only consider neighboring clusters, ignoring global information, which can lead to some defects, such as incomplete coreference prediction and missed clusters. In natural language processing, some reference relationships may involve multiple parts of a document, not just adjacent sentences or paragraphs.

1.1 Study of This Paper

Based on the existing model's description and the identified problems, this paper studies the key issues and their solutions in document-level coreference resolution tasks. By conducting in-depth research on these key issues, a better understanding can be gained of the challenges faced in document-level coreference resolution, laying the foundation for designing more efficient and accurate algorithms. The following describes the research content of this paper. Firstly, syntactic and mention type constraints can be used to filter out some of the more correct potential mentions. Secondly, in the absence of external knowledge, semantics, syntax, and reference types are crucial in mention identification. Then, in coreference prediction, semantics and mention type knowledge help to more accurately judge the similarity between mentions. In coreference prediction, mentions and antecedents should refer to each other. In long-text coreference resolution, distance is no longer a key factor and should not be used as a key constraint in mention and antecedent representation.

Based on the above research, this paper proposes a symmetry-based approach that combines syntactic, semantic, and mention type knowledge to better address the limitations of current models. We conduct experiments on coreference resolution tasks using the OntoNotes 5.0 dataset. To verify the effectiveness of our proposed method, we conduct other related experiments, including a transfer experiment used to verify the symmetry, and an ablation experiment. The main contributions of this paper are summarized as follows:

1) We propose a mention representation method that utilizes syntax, semantics, and mention type information to better represent mentions.
2) We introduce a coreference representation method that leverages semantics and mention type information to better represent the similarity features between mentions.
3) We present a symmetry-based loss function for coreference resolution, which enhances task-specific features.
4) We evaluate our method on the OntoNotes 5.0 dataset and demonstrate that it is an effective approach for coreference resolution.
5) We conduct transfer learning experiments on multiple models and show that the use of symmetry-based methods can improve coreference resolution performance.

2 Related Work

2.1 End-To-End Model

The task of coreference resolution [1,3] involves clustering mentions referring to the same entity. Traditional methods typically involve designing hand-crafted features to detect entity mentions, followed by using machine learning algorithms or traditional deep learning models to predict their relationships. However, end-to-end coreference resolution models address this problem by learning directly from raw text. [4] first applied the end-to-end model to the task of coreference resolution. This model studies end-to-end to perform coreference resolution, roughly described as (1) whether span i is a mention, (2) The likelihood of span j to be a mention, and (3) the possibility that j is an antecedent of i.

$$\mathrm{s(i,j)} = \begin{cases} 0 & j = \epsilon \\ s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j) & j \neq \epsilon \end{cases} \tag{1}$$

Here, $s_{\mathrm{m}}(i)$ is a univariate fraction for span i, indicating the score that span i is recognized as a mention. Similarly, $s_{\mathrm{m}}(i)$ represents the same score, for span j. Meanwhile, $s_{\mathrm{a}}(i,j)$ is a pairwise score for span j and span i, indicating the score that span j serves as the antecedent of span i. $s_{\mathrm{m}}(i), s_{\mathrm{a}}(i,j)$ is calculated as:

$$s_{\mathrm{m}}(i) = \boldsymbol{w}_{\mathrm{m}} \cdot \mathrm{FFNN}_{\mathrm{m}}\left(\boldsymbol{g}_i\right) \tag{2}$$

$$s_{\mathrm{a}}(i,j) = \boldsymbol{w}_{\mathrm{a}} \cdot \mathrm{FFNN}_{\mathrm{a}}\left(\left[\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \phi(i,j)\right]\right) \tag{3}$$

Here, $\cdot$ denotes dot product, $\circ$ denotes element-wise multiplication, and FFNN represents a feedforward neural network that computes non-linear mapping from input to output vectors. Possible spans are represented as $\boldsymbol{g}_i$, $\boldsymbol{g}_i \circ \boldsymbol{g}_j$ indicates explicit element similarity, and feature vector $\phi(i,j)$ represents distance information between metadata and two spans. The vector representation of each token i is denoted as x_i, and the span representation is obtained by applying Bi-LSTM encoding and an attention mechanism to the embedding x.

$$\boldsymbol{x}_t^* = [\boldsymbol{h}_{t,1}, \boldsymbol{h}_{t,-1}] \tag{4}$$

$$\boldsymbol{g}_i = \left[\boldsymbol{x}^*_{\mathrm{START}(i)}, \boldsymbol{x}^*_{\mathrm{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)\right] \tag{5}$$

Here x_t^* denotes the concatenated output of the bi-directional LSTM, $\hat{\boldsymbol{x}}_i$ represents the weighted sum of word vectors within span i, and the feature vector $\phi(i)$ is used to encode the size of span i.

2.2 Recent Work of End-to-End Model

Many subsequent papers have been based on the principles and ideas presented in [4], which improve coreference resolution from different perspectives. The improvements mainly focus on span representation and coreference representation.

Span representation improvements include pre-trained model representations [8,11,13], graph neighborhood information [9], weighted sub-token representation of token [10], cluster-based representations [13], external knowledge-based representations [14], character information, and other feature enhancements to improve mention representation capability [15].

Coreference representation enhancement encompasses various techniques such as maximum spanning tree prediction clusters [16], cluster-based antecedent prediction [17], bilinear function representation for coreference [10], discourse tree features [12], syntax analysis filter clustering [15], graph structured representation of clustering using GCN (Graph Convolutional Networks [18]), singular-plural clustering [19], memory-efficient cluster updates [11], and external knowledge to enhance coreference representation [6,14].

In recent years, coreference resolution has prioritized memory usage over speed and accuracy. To improve reference and antecedent representation, complex features have been introduced. However, the lack of analysis on key factors affecting coreference resolution has increased model complexity. Due to memory constraints, direct reduction of mention candidates has impacted the effectiveness of mention identification. Additionally, at a local level, factors influencing mention identification have not yet been studied, and the importance of mention similarity in coreference prediction has not been considered. Finally, from a global perspective, the symmetry of mentions and their antecedents within the same predicted cluster has been overlooked.

Upon these improved models, we propose a symmetric-based coreference resolution model that avoids the addition of overly complex features to enhance representation. Which combines information of semantics, syntax, and mention type.

3 Our Model

3.1 End-To-End Model

In order to overcome the limitations of S2E [5], which solely relies on semantics to calculate the probability of span clustering and may remove many correct mention candidates through pruning operations, we propose a symmetric-based coreference resolution model. This model takes into account both local and global perspectives in its optimization process, breaking the assumption made by traditional co-reference prediction models which possible mentions should refer to antecedents appearing before them. In this paper, we will provide detailed discussions on the following aspects of our proposed model. The architecture diagram for the model is presented in Fig. 1.

3.2 Input

The input of document is $X = \{t_1, t_2, \ldots, t_n\}$, the word i in document is represented as t_i. The syntactic constraint type for word i is expressed as $\mathbf{x}_i^{con}$. The mention type constraint for word i is expressed as $\mathbf{x}_i^{ent}$. We encode the input and obtain the embedding vector representation of X through Longformer:

$$X^* = Encoder(X) \tag{6}$$

3.3 Local

We add syntactic and mention type constraints x_i^{con}, x_i^{ent} to the embedding vector X^*:

$$X^* = [xi^*; \mathbf{x}_i^{con}; \mathbf{x}_i^{ent}] \tag{7}$$

Mention Representation. To improve the mention representation, we obtain the mention representation of $[m^s, m^e]$ based on Eq. 8 [5], m^s, m^e is the representation of start and the end word of the mention, respectively. Then, we

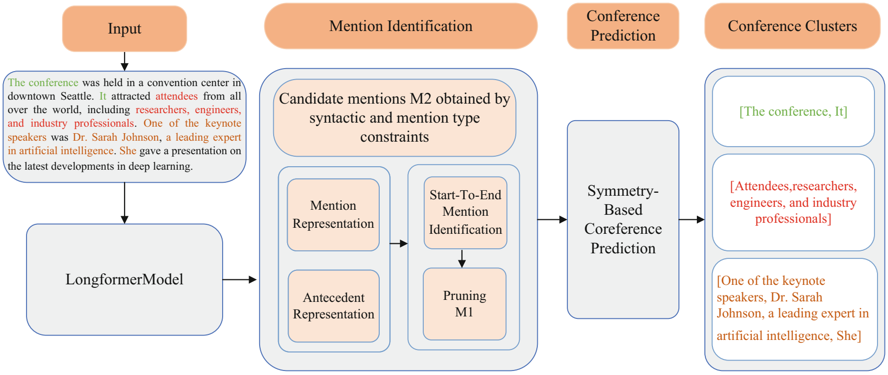

Input
Mention Identification
Conference Prediction
Conference Clusters
The conference was held in a convention center in downtown Seattle. It attracted attendees from all over the world, including researchers, engineers, and industry professionals. One of the keynote speakers was Dr. Sarah Johnson, a leading expert in artificial intelligence. She gave a presentation on the latest developments in deep learning.
LongformerModel
Candidate mentions M2 obtained by syntactic and mention type constraints
Mention Representation
Antecedent Representation
Start-To-End Mention Identification
Pruning M1
Symmetry-Based Coreference Prediction
[The conference, It]
[Attendees,researchers, engineers, and industry professionals]
[One of the keynote speakers, Dr. Sarah Johnson, a leading expert in artificial intelligence, She]

Fig. 1. The overall architecture of our SymCoNLL model. The input channel is sent to module 3.2 for encoding to obtain the semantic embedding representation. Local module 3.3 first adds syntax and mention type representation to get new embedding, uses head and tail word of a span to represent mention and predicts the score of mention, then performs pruning to get potential entity set M1, and gets possible entity set M2 through syntactic and mention type constraints. Then the global module 3.4 uses the set of (M1+M2) for coreference prediction to obtain coreference clusters.

incorporate syntax type constraints x_i^{con} to perform feature enhancement for mention using Eq. 9. Finally, we calculate the representation score for the mention according to Eq. 10 [5].

$$\mathbf{m}^s = \mathbf{GeLU}(\mathbf{W1}_m^s \mathbf{x}^*) \quad \mathbf{m}^e = \mathbf{GeLU}(\mathbf{W1}_m^e \mathbf{x}^*) \tag{8}$$

$$\mathbf{m}^s = [\mathbf{m}^s; \mathbf{x}_i^{con}] \quad \mathbf{m}^e = [\mathbf{m}^e; \mathbf{x}_i^{con}] \tag{9}$$

$$f_m(m) = \mathbf{v}_s \cdot \mathbf{m}_{p_s}^s + \mathbf{v}_e \cdot \mathbf{m}_{p_e}^e + \mathbf{m}_{p_s}^s \cdot \mathbf{H}_m \cdot \mathbf{m}_{p_e}^e \tag{10}$$

Here **GeLU** is an activation function, the matrixs $W1_m^e$, $W1_m^e$ are the trainable parameters of mention representation, the vectors v_s, v_e and the matrix H_m are the trainable parameters of our mention scoring function $f_m(m)$.

Mention Candidates. To generate the final list of possible mentions, we first calculate $f_m(m) > \alpha$ to obtain a set of potential mentions, denoted as $M1$. Here, α represents a hyperparameter that has been predetermined. Next, we obtain another set of potential mentions $M2$, based on syntax and type constraints. Combining $M1$ and $M2$, we arrive at the final set of mention candidates, denoted as M, where $M = M1 + M2$.

Antecedent Representation. Similar to the mention representation, we use X^* to get the antecedent representation $[a^s, a^e]$ using Eq. 11 [5]. Then, we add mention type constraints x_i^{ent} to a to get antecedent representation for enhancing features according to Eq. 12.

Finally, we calculate antecedent representation score by Eq. 13 [5]:

$$\mathbf{a}^s = \mathbf{GeLU}(\mathbf{W2}_m^s \mathbf{x}^*) \quad \mathbf{a}^e = \mathbf{GeLU}(\mathbf{W2}_m^e \mathbf{x}^*) \tag{11}$$

$$\mathbf{a}^s = [\mathbf{a}^s; \mathbf{x}_i^{ent}] \quad \mathbf{a}^e = [\mathbf{a}^e; \mathbf{x}_i^{ent}] \tag{12}$$

$$f_a(c,q) = \mathbf{a}_{c_s}^s \cdot \mathbf{B}_a^{ss} \cdot \mathbf{a}_{q_s}^s + \mathbf{a}_{c_s}^s \cdot \mathbf{B}_a^{se} \cdot \mathbf{a}_{q_e}^e + \mathbf{a}_{c_e}^e \cdot \mathbf{B}_a^{es} \cdot \mathbf{a}_{q_s}^s + \mathbf{a}_{c_e}^e \cdot \mathbf{B}_a^{ee} \cdot \mathbf{a}_{q_e}^e \tag{13}$$

The matrixs $W2_m^s$, $W2_m^e$ are the trainable parameters of antecedent representation. Here $a_{c_s}^s, a_{c_e}^e$ are the start and end representations of the antecedent, respectively. Similarly, $a_{q_s}^s, a_{q_e}^e$ are the start and end representations of the mention, respectively. The matrixs B_a^{ss}, B_a^{se}, B_a^{es}, B_a^{ee} are the trainable parameters of antecedent scoring function f_a.

3.4 Global

Coreference Prediction. The mention candidates and antecedents mentioned above are utilized for co-reference prediction. Previous approaches to coreference prediction have only considered one direction, where the mention candidate points to the antecedent [5].

$$f(c,q) = f_m(m) + f_a(c,q) \tag{14}$$

In consideration of the symmetry between mentions and antecedents pointing to each other, we also consider antecedents pointing to mentions.

$$f(q,c) = f_m(m) + f_a(q,c) \tag{15}$$

Final Score. The final score for coreference resolution is:

$$f = (f(c,q) + f(q,c))/2 \tag{16}$$

4 Experiments and Results

4.1 Tasks

We conducted experiments on two tasks: mention recognition and coreference prediction. Mention recognition involved identifying the boundaries of mention spans to obtain possible mention spans, which were then utilized for co-reference prediction.

Dataset. We perform training and evaluation on the document-level English OntoNotes 5.0 dataset [20], which consists of 2802, 343 and 348 documents in the training, development and test data sets.

Evaluation Metrics. For coreference resolution, we use coreference scorer [21] to evaluate, is the unweighted average of the F1 scores of MUC [22], B^3 [23] and CEAF$_{\phi_4}$ [24]. MUC [22] focuses on the link between mentions and antecedents of the data in coreference prediction. The drawback is that there is no way to measure the performance of the predicted singleton entity. B^3 [23] can overcome the shortcomings of MUC, which focuses on entities. CEAF$_{\phi_4}$ [24] focuses more on the similarity between entities. For mention identification, mention identification results are represented by F1 scores. If a mention exactly matches the gold mention entry on the boundary, the mention entry is considered to be correctly detected.

Experimental Detail. We utilize the Longformer-Large [25] as our underlying pre-training model because of its capability to handle lengthy documents without the need for sliding windows or truncation. The hyperparameters of the model and the optimizer are consistent with the original paper [5], expect epoch is 250 and α is 0.6. Our experiments were focused on two tasks: mention recognition and coreference prediction.

4.2 Baseline

In this paper, a symmetry-based coreference resolution model is proposed for the first time, so its results are compared with existing models of coreference resolution:

G2GT reduced and G2GT overlap [26]: it modeled coreference links in a graph structure where nodes are tags in text and the edges represent the relationship between them.

U-MEM*+SpanBERT [27]: it proposed a memory-enhanced neural network, which only tracks a limited number of entities at a time, thus ensuring linear memory control of document length.

SpanBERT [28]: this work used SpanBERT to better represent and predict the span of text, and is used for coreference resolution tasks.

SpanBERT + CM [27]: its an end-to-end coreference resolution system, and implemented along with four HOI methods, including antecedents, entity equalization, span clustering, and cluster merging.

S2E [5]: it introduced a lightweight end-to-end coreference resolution model, which eliminates the reliance on span representation, manual features, and heuristics.

4.3 Results

The experimental comparison results between baseline models and our proposed model are shown in Table 1. Compared with baselines, our proposed symmetric-based method, which integrates syntactic, semantic, and mention type knowledge, outperforms other methods by a significant margin, particularly in terms of F1 and precision. Furthermore, multiple evaluation benchmarks, including MUC, B^3, and CEAF$_{\phi_4}$, assess the coreference resolution task from various perspectives. The accuracy and F1 on MUC are notably enhanced, indicating that the model has made considerable progress in link prediction for the coreference resolution task. On B^3, the F1 is significantly improved, suggesting that the model has made remarkable strides in mention identification. On CEAF$_{\phi_4}$, the accuracy and F1 are also substantially improved, demonstrating that the model has significantly boosted the accuracy of similarity between mentions in the coreference resolution task. Overall, experiments confirm that the proposed symmetric-based method enhances the coreference resolution task from multiple angles, including mention identification, similarity between mentions, and links between mentions and antecedents, thereby highlighting the effectiveness of this method.

Table 1. Coreference resolution performance of different models on the test set of English OntoNotes 5.0 dataset.

Model	MUC			B^3			CEAF$_{\phi_4}$			CoNLL
	P	R	F1	P	R	F1	P	R	F1	F1
G2GT reduced	85.9	**86.0**	85.9	79.3	**79.4**	79.3	76.4	**75.9**	76.1	80.5
G2GT overlap	85.8	84.9	85.3	78.7	78.0	78.3	76.4	74.5	75.4	79.7
U-MEM+SpanBERT	*	*	*	*	*	*	*	*	*	79.6
SpanBERT	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
SpanBERT+CM	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
S2E	86.5	85.1	85.8	**80.3**	77.9	79.1	76.8	75.4	76.1	80.3
SymCoNLL	**86.6**	85.6	**86.1**	79.9	78.9	**79.4**	**77.9**	75.1	**76.5**	**80.7**

4.4 Ablation Experiments

For the added syntactic features, mention type features, and symmetry loss functions, an ablation experiment is conducted to study the effect of each feature on task efficiency. ‘I’ means to combine syntax and mention type knowledge in the input for embedding representation from Eq. 7, ‘M’ means to add syntax type knowledge for mention representation from Eq. 9, ‘A’ represents the addition of mention type knowledge for antecedent representation is derived from Eq. 12, and ‘sym’ represents the symmetry-based loss function from Eq. 16. The experimental results are shown in Table 2.

The experimental results demonstrate ‘sym’, ‘I’, ‘M’, ‘A’ can improve the efficiency of the model from different degrees on the two tasks. Incorporating symmetry can effectively enhance the efficiency of Mention recognition. Additionally, adding a combination of syntax, semantics and Mention type knowledge can improve the effectiveness of coreference prediction. These features must be combined and adjusted appropriately, and the current implementation is not entirely compatible with both experiments simultaneously. The introduction of symmetry can increase the likelihood of recognizing similar mentions concurrently, while the inclusion of I+M+A enhances the feature representation capacity of mentions and better captures the similarity characteristics of mentions within the same cluster.

4.5 Transfer Experiments

We carried out transfer experiments based on symmetry on two models wl-coref and S2E, then verified the usefulness of symmetry in coreference resolution. The experimental results are shown in Table 3. Compared to wl-coref and S2E,

Table 2. Ablation experiments on the test set of the English OntoNotes5.0 dataset performance on two tasks.

Model	Mention Recognition			Coreference Prediction		
	P	R	F1	P	R	F1
S2E+sym	**89.7**	87.9	**88.8**	**82.1**	79.2	80.6
S2E+M+A+sym	89.6	87.8	88.7	81.5	79.6	80.5
S2E+I+M+A+sym	89.4	**87.9**	88.6	81.4	**79.9**	**80.7**

the study results indicate that the addition of a symmetry-based loss function can effectively improve the F1 value. Properly breaking the unidirectional and sequential nature between mention and antecedent, while adding symmetry can enhance the internal connections among mentions within the same cluster. This approach can better represent the coreference resolution task and significantly improve the efficiency of the coreference resolution model.

Table 3. Performance of transfer experiments on the test set of English OntoNotes5.0 dataset.

Model	MUC			B^3			CEAF$_{\phi_4}$			CoNLL
	P	R	F1	P	R	F1	P	R	F1	F1
wl-coref	84.6	**87.8**	86.1	77.0	**82.2**	79.5	75.6	**76.8**	76.2	80.6
S2E	86.5	85.1	85.8	**80.3**	77.9	79.1	76.8	75.4	76.1	80.3
wl-coref+sym	85.2	87.1	86.2	78.1	81.4	**79.7**	76.2	**76.8**	**76.5**	**80.8**
S2E+sym	**87.0**	85.6	**86.3**	**80.3**	78.6	79.5	**77.6**	74.89	76.2	**80.6**

5 Conclusion

We propose SymCoNLL, a lightweight optimization scheme on the end-to-end coreference resolution model. It combines semantics, syntax and mention type knowledge features to optimize mention identification, and enhances the similarity comparison between mentions. Enhancing the internal connection of mentions in the same cluster based on symmetry aims to improve coreference prediction. It aims to exploit the key factors that affect document-level coreference resolution to improve the overall performance of the task. Our optimization scheme is experimented on multiple baseline models and is competitive while being simple and more efficient. The significance of this study is to deeply understand the task of coreference resolution and provide a positive direction for improving the effect of document-level knowledge extraction tasks in the future.

References

1. Clark, K., Manning, C.D.: Improving coreference resolution by learning entity-level distributed representations. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, August 2016, pp. 643–653. Association for Computational Linguistics (2016)
2. Elango, P.: Coreference Resolution: A Survey, p. 12. University of Wisconsin, Madison (2005)
3. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: CoNLL-2012 shared task: modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL-Shared Task, pp. 1–40 (2012)
4. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, September 2017, pp. 188–197. Association for Computational Linguistics (2017)
5. Kirstain, Y., Ram, O., Levy, O.: Coreference resolution without span representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 14–19 (2021)
6. Zhang, H., Song, Y., Song, Y.: Incorporating context and external knowledge for pronoun coreference resolution. arXiv preprint arXiv:1905.10238 (2019)
7. Lu, J., Ng, V.: Constrained multi-task learning for event coreference resolution. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, June 2021, pp. 4504–4514. Association for Computational Linguistics (2021)
8. Joshi, M., Levy, O., Weld, D.S., Zettlemoyer, L.: BERT for coreference resolution: baselines and analysis. arXiv preprint arXiv:1908.09091 (2019)
9. Fu, Q., Song, L., Du, W., Zhang, Y.: End-to-end AMR coreference resolution. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, August 2021, pp. 4204–4214. Association for Computational Linguistics (2021)
10. Dobrovolskii, V.: Word-level coreference resolution. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, November 2021, pp. 7670–7675. Association for Computational Linguistics (2021)
11. Xia, P., Sedoc, J., Van Durme, B.: Incremental neural coreference resolution in constant memory. arXiv preprint arXiv:2005.00128 (2020)
12. Khosla, S., Fiacco, J., Rosé, C.: Evaluating the impact of a hierarchical discourse representation on entity coreference resolution performance. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, June 2021, pp. 1645–1651. Association for Computational Linguistics (2021)
13. Kantor, B., Globerson, A.: Coreference resolution with entity equalization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 673–677 (2019)
14. Zhang, H., Song, Y., Song, Y., Yu, D.: Knowledge-aware pronoun coreference resolution. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019, pp. 867–876. Association for Computational Linguistics (2019)

15. Fang, K., Jian, F.: Incorporating structural information for better coreference resolution. In: Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI 2019 (2019)
16. Zaporojets, K., Deleu, J., Jiang, Y., Demeester, T., Develder, C.: Towards consistent document-level entity linking: joint models for entity linking and coreference resolution. arXiv preprint arXiv:2108.13530 (2021)
17. Yuan, M., Xia, P., May, C., Van Durme, B., Boyd-Graber, J.: Adapting coreference resolution models through active learning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, pp. 7533–7549. Association for Computational Linguistics
18. Meng, Y., Rumshisky, A.: Triad-based neural network for coreference resolution. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics, pp. 35–43
19. Zhou, E., Choi, J.D.: They exist! Introducing plural mentions to coreference resolution and entity linking. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 24–34 (2018)
20. Pradhan, S., et al.: Towards robust linguistic analysis using ontonotes. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pp. 143–152 (2013)
21. Pradhan, S., Luo, X., Recasens, M., Hovy, E., Ng, V., Strube, M.: Scoring coreference partitions of predicted mentions: a reference implementation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, Maryland, June 2014, pp. 30–35. Association for Computational Linguistics (2014)
22. Vilain, M., Burger, J.D., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, 6–8 November 1995 (1995)
23. Bagga, A., Baldwin, B.: Algorithms for scoring coreference chains. In: The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, vol. 1, pp. 563–566. Citeseer (1998)
24. Luo, X.: On coreference resolution performance metrics. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 25–32 (2005)
25. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
26. Miculicich, L., Henderson, J.: Graph refinement for coreference resolution. In: Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 2022, pp. 2732–2742. Association for Computational Linguistics (2022)
27. Xu, L., Choi, J.D.: Revealing the myth of higher-order inference in coreference resolution. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8527–8533, Online. Association for Computational Linguistics, November 2020
28. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguist. **8**, 64–77 (2020)

What Makes a Charge? Identifying Charge-Discriminative Facts with Legal Elements

Xiyue Luo[1], Wenhan Chao[1], Xian Zhou[2(✉)], Lihong Wang[3],
and Zhunchen Luo[2(✉)]

[1] School of Computer Science and Engineering, Beihang University, Beijing, China
{luoxiyue,chaowenhan}@buaa.edu.cn
[2] Information Research Center of Military Science, PLA Academy of Military Science, Beijing, China
zhouxian@alumni.sjtu.edu.cn, zhunchenluo@gmail.com
[3] National Computer Network Emergency Response Technical Team Coordination Center of China, Beijing, China
wlh@isc.org.cn

Abstract. Over the past few years, there has been a significant surge in the use of deep learning models for automated charge identification tasks, yielding impressive results. However, these models still face several limitations when it comes to explaining the rationale behind charge predictions. Recent research has sought to enhance interpretability by extracting discriminative features from factual descriptions. Nevertheless, such studies often neglect a thorough examination of the corresponding legal provisions. To tackle this issue, the objective of this paper is to establish a fine-grained connection between charges and factual components, specifically identifying charge-discriminative facts that encompass essential legal elements. Several challenges need to be addressed, including the presence of noisy sentences, imbalanced data, and the need for distinct legal elements corresponding to different charges. To tackle these challenges, this paper reframes the task as a few-shot text classification problem and introduces a meta-learning framework that integrates legal element information through a prototypical network. The experimental results demonstrate the effectiveness of this approach in improving charge prediction accuracy and establishing meaningful associations between charges and factual evidences.

Keywords: Charge-discriminative fact identification · Few-shot text classification · Meta-learning framework

1 Introduction

In recent years, there has been a growing interest in the legal field towards the development of machine learning models for automatic charge identification (ACI), also referred to as charge prediction [7,13,18]. The primary goal of ACI is

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 298–310, 2023.
https://doi.org/10.1007/978-3-031-44693-1_24

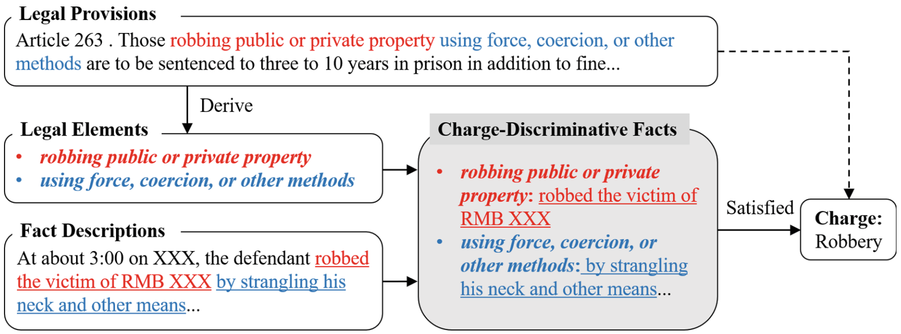

Legal Provisions
Article 263 . Those robbing public or private property using force, coercion, or other methods are to be sentenced to three to 10 years in prison in addition to fine...
Derive
Legal Elements
• robbing public or private property
• using force, coercion, or other methods
Fact Descriptions
At about 3:00 on XXX, the defendant robbed the victim of RMB XXX by strangling his neck and other means...
Charge-Discriminative Facts
• robbing public or private property: robbed the victim of RMB XXX
• using force, coercion, or other methods: by strangling his neck and other means...
Satisfied
Charge: Robbery

Fig. 1. An example of identifying charge-discriminative facts with legal elements.

to automatically predict the charge(s) that could be filed in a case, given the available fact description and legal provisions, using machine learning models, particularly attention-based deep learning models [13]. While these models have demonstrated impressive performance in automatic charge identification, there is still little understanding of how the predicted charges are generated from the perspective of legal regulations and principles. To address this issue, recent studies have focused on improving the interpretability of ACI by extracting discriminative features, such as legal attributes and key events, from fact descriptions [3,5,6,10]. These approaches have been shown to effectively explain which sentences in the fact descriptions are relevant to the predicted charges. However, the extracted features are often based on expert knowledge or event information, lacking a detailed examination of the legal provisions associated with the charges.

In fact, in the criminal justice system, charging an individual or group for a particular case involves a careful analysis of the evidence presented. Basically, the judge must identify and prove that the defendant violated a specific statute or law and that the elements of the crime are satisfied beyond a reasonable doubt. This necessitates a detailed analysis of the relevant legal provisions and their application in the case's specific circumstances. Ultimately, the decision to convict a defendant needs to determine whether the facts of the case fully meet the elements of the charge in accordance with the relevant legal provisions, while ensuring that justice is delivered fairly and impartially.

To gain a better understanding of how charges are convicted in accordance with legal provisions, the aim of this paper is to establish a fine-grained correspondence between charges and fact pieces, that is, *to identify charge-discriminative facts with legal elements from fact descriptions,* as illustrated in Fig. 1. Specifically, our goal is to recognize key factual information that can distinguish different charges, and link them to corresponding legal elements such as criminal intent, object, etc. However, accomplishing this task is not easy as it faces significant difficulties. These difficulties include: (1) the 'noisy' sentences in the fact descriptions that do not indicate any charge [13]; (2) the imbalanced nature of the charge case data, leading to an inferior performance on the rare charges [6]; (3) the need for distinct legal elements to justify different charges, which makes end-to-end methods with simple designs inappropriate.

To tackle these difficulties, we reframe the task into a few-shot text classification problem and devise a meta-learning framework to handle the justification of different legal elements required for different charges. Specifically, we construct a fine-grained charge-fact correspondence dataset, where pieces of fact will be classified as one or more legal elements of a charge based on an inputted support set. For the few-shot learning problem, we employ a prototypical network architecture to learn a non-linear mapping of the input into an embedding space and calculate the prototype representation of each legal element as the mean of its support set in the embedding space. Afterwards, by selecting the prototypical category that exhibits the closest proximity to the text of the fact, we can obtain the predicted element corresponding to the fact. Additionally, we propose a legal element enhanced meta-learning framework, which integrates label information into the prototypical network to improve prototype representations. By utilizing meta-learning techniques, we enhance the model's ability to generalize the correspondence between facts and legal elements so that the model can classify new classes with only a small amount of labeled support data.

We summarize the contributions of this paper as follows: (1) We present a novel task called charge-discriminative fact identification, which focuses on establishing fine-grained charge-fact correspondences with legal elements. In order to address this task, we construct a manually annotated dataset comprising facts related to nine charges. (2) To tackle the imbalanced issue in the proposed task, we transform it into a few-shot text classification problem. Additionally, we propose a meta-learning framework enhanced with legal elements to effectively address the challenge. (3) We conduct extensive experiments on the dataset to evaluate the performance of our framework. The results demonstrate the superiority of our approach. Particularly, the experiments conducted for charge prediction emphasize the efficacy of the extracted charge-discriminative facts in improving the accuracy of the predictions.

2 Related Work

2.1 Charge-Related Fact Extraction

ACI aims to predict charges that may be brought against a defendant in a case based on the given fact description and legal provisions. Since the length of a single fact is relatively long, usually reaching 300–400 words [17], guiding the model to pay attention to the critical information in the facts can improve the performance and interpretability of ACI.

Part of the existing work has noticed this and launched research. Jiang et al. [2] used the attention mechanism to extract essential fact information during the charge prediction task to improve the interpretability of the predictions. Ge et al. [5] and Paul et al. [13] analyzed facts fine-grained by learning the legal provisions and the sentence-level facts correspondence, but they did not further analyze the content and elements of legal provisions. Feng et al. [3] defined a hierarchical event structure for legal cases and extracted the fine-grained event information from the facts to improve legal judgment prediction tasks. But these event details are elements of general criminal events and cannot explain the predicted results from a legal perspective. Hu et al. [6] and Lyu et al. [10] both defined a set of attributes based on experience to obtain a better fact representation. However, the attributes and elements in these works are not directly related to the sentencing dimensions emphasized in the legal provisions.

These works used the same elements for different charges to guide the identification of facts, clearly distinct from the actual judging process. It is obvious that in the practical judging process, analyzing different types of cases requires the use of different legal provisions. So, we believe that distinct fact identification tasks should be designed to suit different charges, which has not yet been studied in the existing work we have researched.

2.2 Meta-learning

Meta-learning, also known as learning to learn, involves enhancing a learning algorithm over multiple learning episodes. Its primary objective is to enable the algorithm to solve new, unseen tasks with minimal examples by leveraging prior knowledge acquired through previous learning experiences.

There are three main approaches to meta-learning: The first is model-based meta-learning, in which a meta-model is used to update or predict the weights of a task-specific mode [11]. The second approach is metric-based meta-learning, which is similar to nearest neighbors algorithms [14,15]. These methods use an embedding space or a metric that is learned during meta-training and can be used during meta-testing to embed both new categories' few support examples and queries. The third approach is optimization-based meta-learning, which explicitly learns an update rule or weight initialization that facilitates rapid learning during meta-testing. Finn et al. [4] used ordinary gradient descent in the learner and meta-learned the initialization weights, and this idea was further extended by Nichol et al. [12] to propose Reptile, an algorithm that does not require a training-test split for each task. Overall, the goal of meta-learning is to enable machine learning algorithms to quickly adapt to new tasks or environments by utilizing previously learned knowledge.

3 Problem Definition

The definitions of the terms involved in our task are as follows:

Legal element is the key factor that can impact the charges and sentencing decisions. The legal elements of each charge are obtained by analyzing the corresponding legal provisions and deconstructing them.

Charge-discriminative fact is the critical information that can distinguish various charges and is discriminative in the corresponding element. The charge-discriminative facts are identified base on the corresponding legal elements. These facts, together with the legal elements and provisions, can be utilized to predict charges.

The task is defined as follows: Charge c_i has charge-discriminative facts identification task T_i, and the corresponding dataset is $D_i = (x_{ij}, e_{ij}, r_{ij})$. x_{ij}, e_{ij}, r_{ij} represent a sentence-level fact, a legal element corresponding to the charge c_i, and the relationship between the fact and the legal element, respectively. For each task T_i, our goal is to learn a classification function f_i, which is used to predict whether the fact x_{ij} is discriminative in the element e_{ij}. $P(T)$ contains a set of tasks, and the ultimate goal of our meta-learning framework is to find a model f that can fit into all these tasks. We define this task as a few-shot text classification task. In the K-shot scenario, the model uses K samples to train the classification function f_i based on the new task T_i.

4 Dataset

4.1 Construction Process

Charge Selection. Different charges correspond to different legal element classification tasks. We hope that the meta-model can learn about the common parts of knowledge in all charges and be used for predicting new charges, whether they are similar or dissimilar. Therefore, we selected several similar charges, such as theft and fraud, and traffic accidents, which were not strongly related to other charges. Finally, nine charges were selected and annotated.

Dataset Construction. A law school graduate helped us analyze the legal provisions into legal elements which are elements that can determine charges and affect sentencing. For example, the use of violence and serious injury are the corresponding elements of these two factors of robbery. For each charge, we identified 8–13 legal elements. It is worth noting that the corresponding elements are different for different charges.

Then we construct the fact set. All case descriptions in the dataset we constructed come from public criminal judgment documents in China Judgements Online (CJO)[1]. We randomly selected 30 judgment documents for each chosen charge and divided them into sentence units. Finally, we wrote a set of simple regular screening rules to filter out the sentences that are irrelevant to the case. The remaining sentences of the documents constituted the fact set.

Charge-Discriminative Facts Annotation. Three computer graduate students completed this annotation under the guidance of the law school graduate. The annotated contents are the charge-discriminative facts from the fact set with the above-mentioned legal elements. Charge-discriminative facts mean that this fact can demonstrate the element. For example, strangling his neck and other means is discriminative in determining the use of violence.

The detailed statistics on our dataset are shown in Table 1, where OE, FI, and TA represent official embezzlement, false emphasis, and traffic accident,

[1] http://wenshu.court.gov.cn.

Table 1. Statistics of the dataset.

Charges	robbery	theft	fraud	OE	injury	rape	kidnap	FI	TA
#legal elements	13	13	10	9	8	10	11	10	13
#facts	173	363	227	167	143	154	155	207	201
#avg. facts	13.3	27.9	22.7	18.6	17.9	15.4	14.1	20.7	15.5

respectively. We annotated around ten legal elements for each charge. Each legal element has about 15 labeled data, which satisfies the requirements for few-shot classification tasks. Moreover, it is very convenient to use the framework in a new charge because only a small amount of data needs to be manually annotated.

4.2 Comparison with Previous Datasets

There are already many datasets in the field of intelligent justice. CAIL2018 [17] is a commonly used dataset which collected 2,676,075 criminal cases from CJO. For a given fact, CAIL contains the corresponding law, charge and sentence. ECHR [1] is a similar dataset in English. But these datasets are extracted with regular expressions rather than manually annotated and do not contain fine-grained information.

Feng et al. [3] and Ge et al. [5] presented their datasets in their work, respectively. These datasets are manually annotated with fine-grained fact information. However, these datasets do not have a fine-grained analysis of the legal provisions corresponding to the charge, and the annotated information cannot explain the charge according to the content of the law. Therefore, we annotated and made this dataset publicly available.

5 Model

China's Criminal Law outlines 483 distinct crimes, with a notable variance in the frequency of cases for each offense [6]. Moreover, new legal regulations are frequently introduced. In 2022, China has enacted 6 laws and amended 10 laws. This necessitates the ability of our framework to acclimate to recently-evolved charges and situational logic rapidly, so we employ the powerful meta-learning framework to quickly adapt to the new given case type and judgment logic.

5.1 Legal Element Enhanced Meta-learning Framework

We propose a legal element enhanced meta-learning framework (LEEMF) to solve this task, as depicted in Fig. 2. The text information of the legal elements is used to enhance the framework's performance, and we make the prototypes additionally represent whether the fact is discriminative in the element.

We use the pre-trained language model (PLM) to represent text and the text information of legal elements to enhance it. The input format is "[CLS] fact [SEP] legal element [SEP]" and the hidden-state vectors are taken as the text

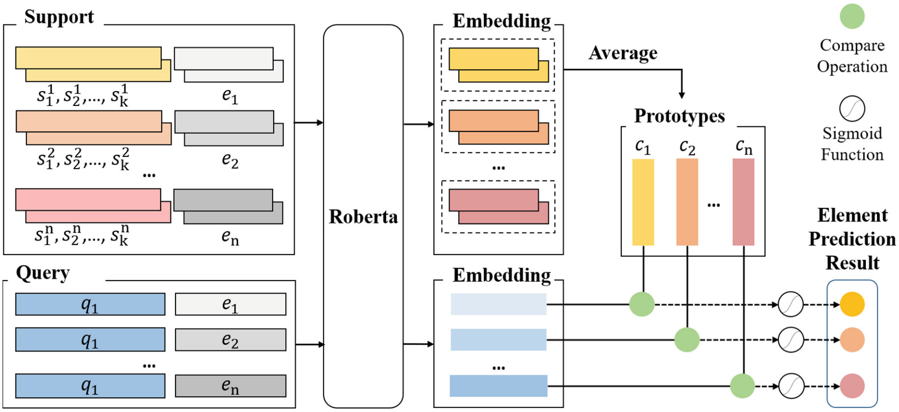


Fig. 2. An overview of our Framework LEEMF in the n-shot k-way setting, where s_j^i represents the fact text j classified into element i, e_i represents the text of element i, q_i represents the query fact text, and c_k represents the prototype k.

representation r. We denote the element e^i and the sentence-level fact classified into e^i as s_i^j. The embedding function is expressed as f_θ where θ represents the meta-learned parameters.

Then the prototypes of each element are calculated in the Support set. We denote the representation of all data classified into element k as S_k. For label k, Its prototype is computed using representations enhanced with correct legal elements.

$$r_j = f_\theta([s_j^i, e_i]) \tag{1}$$

$$c_k = \frac{1}{|S_k|} \Sigma_{r \in S_k} r \tag{2}$$

We make the charge predictions based on the legal element enhanced representation of the query set and update the model. However, the exact elements of the query q_i are unknown, so we match q_i with all elements in the element set ε. Afterward, the classification of each query fact into a certain element is calculated by the distance from the query representation r_i^k to the corresponding prototype c_k. This distance is passed through a sigmoid function to obtain the classification result of a single element.

$$r_i^k = f_\theta([q_i, e_k]), k \in \varepsilon \tag{3}$$

$$d_i^k = exp(-\|r_i^k - c_k\|^2) \tag{4}$$

$$p(y_i = k \mid x) = Sigmoid(\frac{d_i^k}{\Sigma_{k' \in \varepsilon} d_i^{k'}}) \tag{5}$$

$$L = \frac{1}{|\varepsilon|} \Sigma_{k \in \varepsilon} -(y_i^k \cdot log(y_i^k) + (1 - y_i^k) \cdot log(1 - y_i^k)) \tag{6}$$

Algorithm 1. Meta Optimization

```
INITIALIZE $\theta$
while has not converged do
    for charge $c$ in $C$ do
        SAMPLE $\varepsilon, S, Q$
        for element $k$ in $\varepsilon$ do
            $c_k \leftarrow \frac{1}{|S_k|} \Sigma_{r \in S_k} r$
        end for
        $L \leftarrow$ LOSS$(p, BATCH(Q))$
        $\varepsilon \leftarrow$ UPDATE$(\theta_{t-1}, \frac{\partial L}{\partial \theta_{t-1}})$
    end for
end while
```

5.2 Optimization

The meta-learning algorithm of our framework is similar to that of Lichy et al. [8], which combines Prototypical Networks and Reptile and is shown in Algorithm 1. The module f_θ is first initialized with the PLM parameters. Then, in each training epoch, we sample subsets for each charge c in training charges set C to form the element set ε, Support set S and Query set Q. We calculate the prototype based on all samples in the Support set and update the meta-model with the loss L calculated in the Query set.

6 Experiment and Analysis

6.1 Baselines

We compare the proposed model with the following baselines:

- **MAML** [4] learns to adapt rapidly to new tasks with only a few gradient steps. The model's parameters are explicitly trained with a small amount of training data from a new task will produce good generalization performance.
- **Reptile** [15] works by repeatedly sampling a task, training on it, and moving the initialization towards the trained weights on that task. Unlike MAML, the meta-model weights will be fine-tuned during meta-testing.
- **Prototypical Network** [14] (ProtoNet) is a metric-based meta-learning method for few-shot classification which learns to align query instances with class prototypes.
- **Hierarchical Matching Network** [16] (HMN) fuses both hierarchical structure and semantics of labels to predict correct laws and articles to solve the charge classification problem.

6.2 Experimental Settings

As our dataset comprises nine charges, we utilized the nine-fold cross-validation approach. This entailed utilizing eight data types for training and the remaining

Table 2. Experimental results of 5-way 1-shot, 3-shot and 5-shot on our dataset. P, R, and F1 represent precision, recall, and f1 score, respectively.

Models	1-shot			3-shot			5-shot		
	P	R	F1	P	R	F1	P	R	F1
MAML+LBert	**0.848**	0.635	0.717	0.847	0.752	0.797	0.839	0.774	0.805
Reptile+LBert	0.823	0.587	0.683	**0.858**	0.653	0.739	0.876	0.719	0.784
ProtoNet+LBert	0.781	**0.740**	**0.758**	0.762	0.822	0.787	0.805	0.837	0.819
LEEMF+LBert	0.752	0.680	0.707	0.776	0.816	0.789	0.829	0.808	0.817
MAML+Roberta	0.818	0.703	0.717	0.802	0.776	0.789	0.820	0.779	0.799
Reptile+Roberta	0.738	0.643	0.687	0.750	0.620	0.672	0.816	0.681	0.737
ProtoNet+Roberta	0.741	0.738	0.739	0.760	0.803	0.774	**0.847**	0.798	0.819
LEEMF+Roberta	0.730	0.737	0.734	0.798	**0.834**	**0.811**	0.833	**0.841**	**0.835**

type for testing. The model's performance was then evaluated by averaging the results obtained from the nine testing sets.

In our framework, we mainly utilized two pre-trained language models, namely Legal BERT (LBert) [19] and Roberta [9]. LBert was trained on Bert-base using 6.63 million criminal documents, providing it with extensive legal knowledge.

To demonstrate the utility of charge-discriminative facts in other ACI tasks, we conducted a few-shot charge prediction experiment across the nine defined charges. In each charge prediction task, we trained the framework using only a minimal amount of supervised data to simulate the scenario of low-frequency charge prediction.

6.3 Experimental Results

Comparison of Proposed Approach with Baselines. Table 2 presents the comparison results between LEEMF and the baseline models. It is evident that in 3-shot and 5-shot scenarios, LEEMF outperforms the other baselines, while ProtoNet performs better in the 1-shot scenario. Furthermore, as the number of shots increases, all four models exhibit enhanced performance.

When comparing different pre-trained language models across all three baseline models, Legal Bert (LBert) outperforms Roberta. This could be attributed to LBert's extensive legal knowledge, which is acquired through pre-training with a vast collection of criminal documents. However, in the case of LEEMF, Roberta performs better than LBert due to its superior sentence relationship prediction capabilities, allowing it to more accurately predict the relationship between factual information and legal elements.

To investigate the impact of different ways during training, we conducted experiments with 5-shots 3-ways, 5-ways, and 7-ways. The results are presented in Table 3, and it is apparent that our framework, LEEMF, outperforms the baseline models in all three scenarios. Additionally, we observed that the model's

Table 3. Experimental results of 5-shot 3-way, 5-way and 7-way on our dataset.

Models	3-way			5-way			7-way		
	P	R	F1	P	R	F1	P	R	F1
MAML+LBert	0.813	0.792	0.799	0.839	0.774	0.805	0.843	0.807	0.825
Reptile+LBert	0.826	0.780	0.798	**0.876**	0.719	0.784	0.837	0.722	0.789
ProtoNet+LBert	0.778	**0.837**	0.804	0.805	0.837	0.819	0.794	0.820	0.804
LEEMF+Roberta	**0.831**	0.807	**0.812**	0.833	**0.841**	**0.835**	**0.859**	**0.862**	**0.858**

Table 4. Experimental results of f1 scores on different charges.

Models	robbery	theft	fraud	OE	injury	rape	kidnap	FI	TA
MAML+LBert	0.796	0.715	0.898	0.917	0.875	**0.809**	0.710	0.726	0.827
Reptile+LBert	0.783	0.639	**0.910**	**0.934**	0.848	0.801	0.747	0.533	0.861
ProtoNet+LBert	0.758	0.745	0.886	0.899	0.910	0.741	0.786	0.828	0.818
LEEMF+Roberta	**0.799**	**0.750**	0.868	0.817	**0.923**	0.791	**0.813**	**0.884**	**0.867**

performance improves as the number of ways in training increases. As the number of ways in training increases, the model's performance improves because it learns a greater amount of information during training.

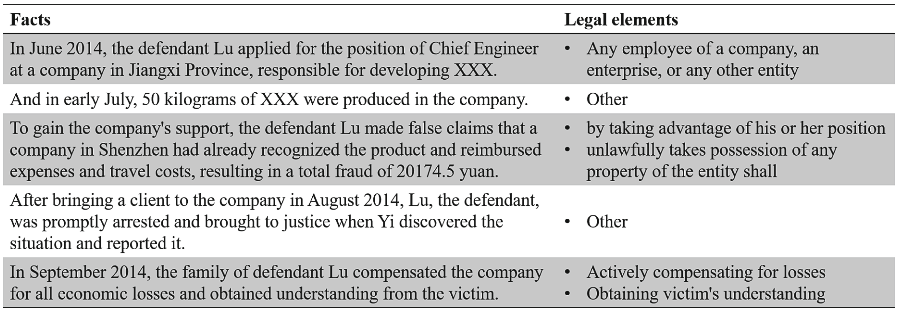
When comparing our frameworks with the baselines, it becomes apparent that LEEMF outperforms the other baselines in all scenarios except for the 5-way 1-shot scenario. This is because when the model is trained with too little data, incorporating legal element information can impede its ability to learn factual information.

Analysis on Different Charges. In order to understand the fact identification performance of the model on different charges, we analyzed the experimental results in 5-shot and 5-way scenarios. The results are shown in Table 4, where OE, FI, and TA represent official embezzlement, false emphasis, and traffic accident, respectively. Among them, theft has the worst prediction effect, possibly due to many unique elements of the crime of stealing bags, such as burglary. The facts corresponding to these elements do not have noticeable semantic features, so the model is difficult to judge accurately. Experiments have proved that for different charges, the identification effect of fact is different, which mainly depends on the elements to which the charge belongs.

Performance on Charge Prediction. To demonstrate that the framework's identified charge-discriminative facts are the critical information that can distinguish various charges, our team designed a framework that utilizes LEEMF to improve the accuracy of charge prediction. The framework contains already trained LEEMF and an ACI model to be trained to predict charges.

Table 5. Experimental results of charge prediction.

Models	5-shot			10-shot			30-shot		
	P	R	F1	P	R	F1	P	R	F1
HMN	0.619	0.578	0.597	0.772	0.753	0.762	0.852	0.837	0.845
HMN+LEEMF	0.692	0.635	0.662	0.802	0.784	0.793	0.877	0.860	0.868
Impv	11.8%	9.86%	10.9%	3.89%	4.12%	4.07%	2.93%	2.75%	2.72%



Facts	**Legal elements**
In June 2014, the defendant Lu applied for the position of Chief Engineer at a company in Jiangxi Province, responsible for developing XXX.	• Any employee of a company, an enterprise, or any other entity
And in early July, 50 kilograms of XXX were produced in the company.	• Other
To gain the company's support, the defendant Lu made false claims that a company in Shenzhen had already recognized the product and reimbursed expenses and travel costs, resulting in a total fraud of 20174.5 yuan.	• by taking advantage of his or her position • unlawfully takes possession of any property of the entity shall
After bringing a client to the company in August 2014, Lu, the defendant, was promptly arrested and brought to justice when Yi discovered the situation and reported it.	• Other
In September 2014, the family of defendant Lu compensated the company for all economic losses and obtained understanding from the victim.	• Actively compensating for losses • Obtaining victim's understanding

Fig. 3. An example of identifying charge-discriminative facts.

The ACI model here can be set as any charge prediction model to prove that LEEMF can widely improve the performance of charge prediction. Our only modification to the ACI model involved extracting the final charge prediction layer representations and fusing them with the element prediction outputs for various charges generated by the meta-model. Finally, we use the fused representation for charge prediction.

A few-shot experiment is designed to evaluate the impact of charge-discriminative facts on the charge prediction task and the result is shown in Table 5. The model was trained on 5, 10, and 30 samples per charge, respectively. Our experimental results demonstrate that the identified charge-discriminative facts can enhance the performance of existing ACI models in predicting low-frequency charges. Furthermore, the improvement is more significant with fewer training samples, with a maximum F1 value improvement of 10.9% achieved in the 5-shot setting.

Case Study. To intuitively illustrate how to identify charge-discriminative facts with legal elements, we present an example of a case description involving official embezzlement. The example, shown in Fig. 3, provides evidence of our framework's ability to accurately identify charge-discriminative facts. These facts can enhance interpretability for ACI tasks and help answer the question of what constitutes a charge in a case description.

7 Conclusion

This paper introduces a novel task focused on identifying charge-discriminative facts enriched with legal elements. We also provide a manually annotated dataset specifically designed for this task. To address this task, we propose a legal element enhanced meta-learning framework which combines the label information of the legal elements with a prototypical network, enabling effective few-shot learning. Experimental results conducted on the constructed dataset validate the efficacy of our proposed method. To promote transparency and facilitate further research, our data and code will be made publicly available on GitHub.

References

1. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European court of human rights: a natural language processing perspective. PeerJ Comput. Sci. **2**, e93 (2016)
2. Chao, W., Jiang, X., Luo, Z., Hu, Y., Ma, W.: Interpretable charge prediction for criminal cases with dynamic rationale attention. J. Artif. Intell. Res. **66**, 743–764 (2019)
3. Feng, Y., Li, C., Ng, V.: Legal judgment prediction via event extraction with constraints. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 648–664 (2022)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
5. Ge, J., Huang, Y., Shen, X., Li, C., Hu, W.: Learning fine-grained fact-article correspondence in legal cases. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3694–3706 (2021)
6. Hu, Z., Li, X., Tu, C., Liu, Z., Sun, M.: Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 487–498 (2018)
7. Le, Y., Zhao, Y., Chen, M., Quan, Z., He, X., Li, K.: Legal charge prediction via bilinear attention network. In: CIKM, pp. 1024–1033 (2022)
8. de Lichy, C., Glaude, H., Campbell, W.: Meta-learning for few-shot named entity recognition. In: Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing, pp. 44–58 (2021)
9. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
10. Lyu, Y., et al.: Improving legal judgment prediction through reinforced criminal element extraction. Inf. Process. Manage. **59**(1), 102780 (2022)
11. Munkhdalai, T., Yu, H.: Meta networks. In: International Conference on Machine Learning, pp. 2554–2563. PMLR (2017)
12. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
13. Paul, S., Goyal, P., Ghosh, S.: Automatic charge identification from facts: a few sentence-level charge annotations is all you need. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1011–1022 (2020)
14. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

15. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
16. Wang, P., Fan, Y., Niu, S., Yang, Z., Zhang, Y., Guo, J.: Hierarchical matching network for crime classification. In: SIGIR, pp. 325–334 (2019)
17. Xiao, C., et al.: CAIL 2018: a large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)
18. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: EMNLP, pp. 3540–3549 (2018)
19. Zhong, H., Zhang, Z., Liu, Z., Sun, M.: Open Chinese language pre-trained model zoo. Technical report (2019). https://github.com/thunlp/openclap

Oral: Machine Learning for NLP

A New Encoder Using Character and Word Feature Fusion for Chinese Math Word Problem Solving

Wenqing Huang and Jing Xiao(✉)

School of Computer Science, South China Normal University, Guangzhou, China
2021023214@m.scnu.edu.cn, xiaojing@scnu.edu.cn

Abstract. Automatically solving math word problems (MWPs) aims at generating corresponding math expressions based on the problem text. Since pre-trained language models can provide rich semantic representation for decoders, they can be used as the encoders for MWP models. However, most Chinese version of pre-trained language models use characters as the basic unit for downstream tasks, thus using them as encoders will lose the information carried by words. In addition, most of current methods ignore the difference between the operators +, ×, and -, ÷, ∧. They neglect to make the model understand the position order of the left and right subtree of operators -, ÷, and ∧ is important. To address the above issues, we propose a Character and Word Fusion Encoder (CWFE) and a negative label loss function in this paper. CWFE can fuse character and word feature information in problem text. Negative label loss prevents the model from incorrectly resolving the positions of operands. Using GTS as the decoder, our model called CWFE-GTS-N outperforms the models HGEN, RPKHS, and E-GTS in the experiments. The accuracy of CWFE-GTS-N reaches 85.9% and 86.4% on two datasets, Math23K and Ape-clean~, respectively. Code is available at https://github.com/SCNU203/CWFE-GTS-N

Keywords: Chinese MWP · Character and Word Feature Fusion · Pre-trained models · Negative Label Loss

1 Introduction

Automatically solving math word problems (MWPs) is a challenging task in natural language processing. It aims at generating correct math expressions and results based on the problem text. In recent years, the use of deep learning models has become mainstream in the field of MWPs, and most of these models consist of an encoder and a decoder. The well-known GTS [1] uses GRU [2] as an encoder and its decoder is a goal-driven tree structure decoder. In terms of decoder, most of the research works after GTS use the tree decoder of GTS as decoders due to its good performance. For encoder, using GRU as an encoder suffers the following problems: ①Ignoring the keywords that appear earlier in the text. ②Lack of external background knowledge. Therefore, there are many

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 313–324, 2023.
https://doi.org/10.1007/978-3-031-44693-1_25

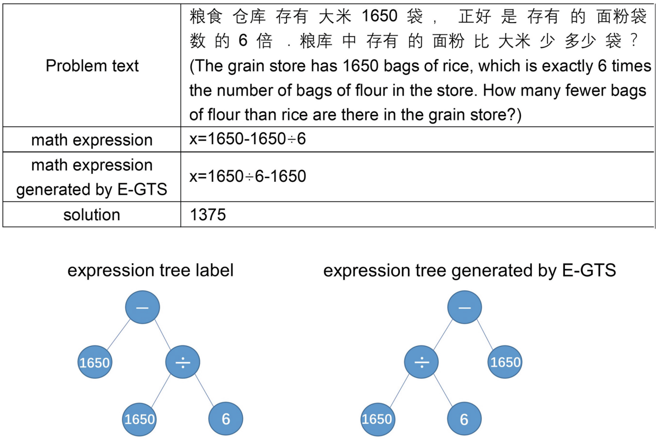
Table 1. A wrong example generated by E-GTS due to the lack of word information.

problem text	一 个 棱 长 为 6 分 米 的 正方体 木块 的 表面积 = 多 少 平方分米 ?(How many square decimeters is the surface area of a cube wood block with a length of 6 decimetres?)
text fed to ELECTRA	一 个 棱 长 为 6 分 米 的 正 方 体 木 块 的 表 面 积 = 多 少 平 方 分 米 .
math expression	x=6*6*6
math expression generated by E-GTS	x=6*6
solution	216

research works focusing on encoder improvements. HGEN [3] proposes a new hierarchical heterogeneous graph encoder to extract more features in the text. RPKHS [4] proposes a hierarchical inference encoder for integrating word-level and sentence-level inference. To advocate research in this direction, in this paper we try to improve the encoder performance and provide more accurate text information to the tree decoder. Pre-trained language models are suitable as encoders for MWPs domain, their advantages over recurrent neural networks such as GRU are as follows: ① The self-attention mechanism enables them to remember keywords that appear earlier in the text. ② Unsupervised training on a large corpus allows them to learn enough external knowledge. Given the above advantages of pre-trained language models, we construct a baseline model called E-GTS which uses the pre-trained language model ELECTRA [5] as the encoder and GTS's decoder as the decoder.

However, pre-trained language models used in Chinese MWPs as encoders only capture the character information in problem text. The lack of word information causes the model to ignore some correct keywords. As shown in the example in Table 1, the original problem text is presented in the first row. The problem's question is to ask the surface area of a cube, but E-GTS generates a math expression for the surface area of a square. It is obvious that E-GTS does not consider "正方(square)" and "体" as a whole of "正方体(cube)". The reason for this situation is that the pre-training language model as an encoder splits all words in Chinese text into characters while losing Chinese word information, as shown in the second row of Table 1. On the other hand, in data preprocessing phase, math expression labels are converted to expression tree labels in a binary tree. However, most of current works neglect to make the model understand that the position order of the left and right subtree of operators -, ÷, and $\wedge$ is important. If a model resolves the position of operands of operators -, ÷, and $\wedge$ incorrectly, it will result in an incorrect solution, as shown in Fig. 1.

To address the above issues, in order to provide the decoder with richer semantic representation, a Character and Word Fusion Encoder (CWFE) is proposed in this paper. Specifically, it integrates boundary feature information for both characters and words in problem text into ELECTRA's semantic representation, effectively fusing the feature information of both Chinese characters and words. What's more, to prevent the model from incorrectly resolving the



Problem text	粮食 仓库 存有 大米 1650 袋 ， 正好 是 存有 的 面粉袋数 的 6 倍 . 粮库 中 存有 的 面粉 比 大米 少 多少 袋 ？ (The grain store has 1650 bags of rice, which is exactly 6 times the number of bags of flour in the store. How many fewer bags of flour than rice are there in the grain store?)
math expression	x=1650-1650÷6
math expression generated by E-GTS	x=1650÷6-1650
solution	1375

Fig. 1. A wrong example generated by E-GTS. The lower left corner is expression tree label converted from the math expression label, and the lower right corner is expression tree generated by E-GTS. Obviously, E-GTS incorrectly resolves the positions of the left and right subtree of operator -.

positions of some operands, such as divisor and dividend, we introduce a negative label loss function. It makes the model to understand that the order of the position of the left and right subtree of operators -, ÷, and ∧ is important. Combining the CWFE, GTS tree decoder, and the negative label loss, the model CWFE-GTS-N is proposed. The main contributions of this paper are summarized as follows:

- We propose CWFE which can provide the decoder with character and word feature information for solving Chinese MWPs.
- We design negative label loss to prevent the model from incorrectly resolving the positions of operands, such as divisor and dividend.
- Experiments show that our CWFE-GTS-N improves 0.8% over E-GTS on dataset Ape-clean~. CWFE-GTS-N improves 7.2%, 2%, and 1.6% over several baseline models of HGEN, RPKHS, and E-GTS on dataset Math23K, respectively.

2 Related Work

The research on automatic MWPs solving can be traced back to the 1960s. Before 2010, researchers developed a series of solution models that rely on rules and patterns in different fields such as algebra, percentage, ratio. These models focus on manual extraction features and belong to semi-automatic solution models such as WORDPRO [6] and ROBUST [7]. And then, Kushman et al. [8] used statistical methods to solve MWPs. This laid the foundation for designing MWPs models using traditional machine learning methods. However, these methods require a lot of manpower to label complex data.

With the development of deep learning and the emergence of large datasets, deep learning solution models have become mainstream in the field of MWPs. The earliest work on solving MWPs using the deep learning model was done by Wang et al. [9]. They proposed an improved sequence-to-sequence model that transforms the problem text into math expressions. Subsequently, Wang et al. [10] used equation normalization to deal with different equivalent expressions, thus enhancing the generalization of the solution model. The GTS authors designed the tree decoder based on the way like human beings think about solving MWPs. After the emergence of GTS, many research works used the tree decoder of GTS as decoders for their models. TSN-MD [11] obtained better results by transferring knowledge to student networks through teacher networks. RPKHS [4] proposed a hierarchical inference encoder for integrating word-level and sentence-level inference. Li et al. [12] used a contrastive learning approach to make the solution model more discriminatory for different equation templates. HGEN [3] proposes a new hierarchical heterogeneous graph encoder to extract more features in the text.

3 Model Description

3.1 Problem Definition and Data Pre-processing

Problem Definition. A math word problem $P = (p_1, p_2, ..., p_j)$ is a sequence of j tokens, where each token can be a character, word, or numeric value. Our goal is to generate the correct math expression based on P.

Data Pre-processing. We split all words in the problem text into characters. The ith numeric value p_k appearing in P is denoted as N_i. For example, in the problem text of Fig. 1, $N_1 = 1650$, $N_2 = 6$. Then all numeric values are replaced with the token NUM. In the end, the tokens CLS and EOS are added at the beginning and the end of the sequence, and the sequence $P^{'}$ of length n is obtained.

The math expression label E_p is a sequence consist of operators (e.g. +, -, ×, ÷, $\wedge$), constant values (e.g. 1, 3.14), and numeric values. We replace all numeric values in E_p with the corresponding number token N_i and convert E_p into the corresponding prefix form T, $T = (y_1, y_2, \ldots, y_m)$. The prior-order traversal of the math binary tree corresponding to E_p is equivalent to T, so we call T the expression tree label.

3.2 Encoder

The overview of our proposed model CWFE-GTS-N is shown in Fig. 2. The encoder CWFE and its components are described in detail in the next four subsections.

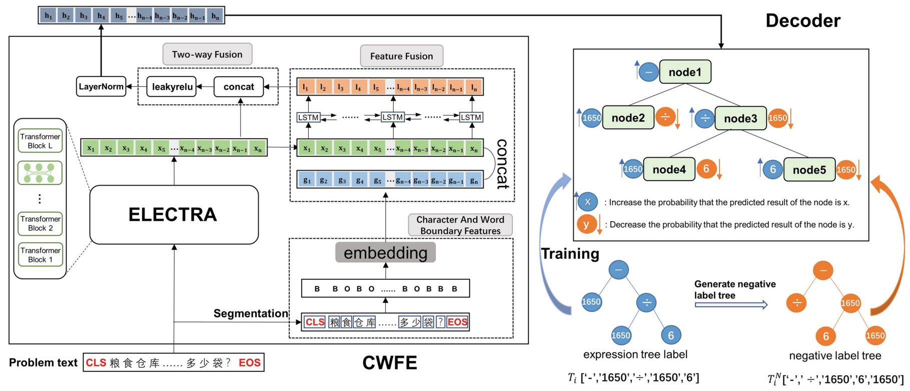

Decoder
Two-way Fusion
Feature Fusion
LayerNorm
leakyrelu
concat
LSTM
concat
Transformer Block L
Transformer Block 2
Transformer Block 1
ELECTRA
Character And Word Boundary Features
embedding
B B O B O ...... B O B B B
Segmentation
CLS 粮食仓库......多少袋？EOS
Problem text
CWFE
node1
node2
node3
node4
node5
1650
6
: Increase the probability that the predicted result of the node is x.
: Decrease the probability that the predicted result of the node is y.
Training
Generate negative label tree
expression tree label
negative label tree
T_i ['-','1650','÷','1650','6']
T_i^N ['-',' ÷','1650','6','1650']

Fig. 2. The overview of CWFE-GTS-N

ELECTRA Language Model. To improve model's understanding of problem text, ELECTRA [5] is chosen as one of the core components of CWFE. Experiments show that the contextual representations learned by ELECTRA substantially outperform the ones learned by BERT given the same model settings [5].

After tokenization, the input sequence $P^{'}$ is converted to $X = (x_1, x_2 \ldots, x_n)$, where each token x_i represents the position of the corresponding token in $P^{'}$ in the input vocabulary. Then the sequence X is input into the ELECTRA language model:

$$X^{'} = ELECTRA(X) \tag{1}$$

where $ELECTRA(\cdot)$ denotes the function of the ELECTRA language model and $X^{'} = (\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_n)$ is the sequence of semantic representations of X computed by ELECTRA.

Character and Word Boundary Feature. This subsection describes the method of obtaining character and word boundary features in problem text. The Chinese text segmentation tool jieba[1] is used to segment the pre-processed problem text and then construct the boundary feature with the following construction rules:

- The first character of each word is marked B.
- Each word is marked with O from the second character until the last character.
- Single character is marked B.

The boundary feature sequence $G = (g_1, g_2, \ldots, g_n)$ is obtained according to the above rules. For example, the boundary feature sequence G corresponding to the sentence "正方体 木块 的 表面积" is "B O O B O B B O O". Then the

[1] github.com/fxsjy/jieba.

embedding vector $\mathbf{g}_k \in \mathbb{R}^d$ is calculated for each element g_k in G by following equation:

$$\mathbf{g}_k = \mathbf{W}_{in}[g_k] \tag{2}$$

where $\mathbf{W}_{in} \in \mathbb{R}^{2\times d}$ is a trainable matrix, d is the output dimension of ELECTRA, $\mathbf{W}_{in}[g_k]$ denotes taking the g_kth row of the matrix $\mathbf{W}_{in}$, element g_k takes only two possible values - 0 or 1 (B or O). Finally, the embedding vector sequence $G^{'} = (\mathbf{g}_1, \mathbf{g}_2 \ldots, \mathbf{g}_n)$ of G is obtained.

Feature Fusion. This section introduces the fusion method of $X^{'}$ and $G^{'}$. First, the dimensions of $X^{'}$ and $G^{'}$ are concatenated to obtain $M = (\mathbf{m}_1, \mathbf{m}_2 \ldots, \mathbf{m}_n)$. Each element $\mathbf{m}_k \in \mathbb{R}^{2d}$ in M is the concatenation of $\mathbf{x}_k$ and $\mathbf{g}_k$. Then, M is fed into a bidirectional LSTM [13] (BiLSTM) network that encodes each $\mathbf{m}_k$ to obtain the corresponding vector $\mathbf{l}_k \in \mathbb{R}^d$. The calculation is as follows:

$$\begin{aligned} \overrightarrow{\mathbf{l}_k} &= LSTM(\overrightarrow{\mathbf{l}_{k-1}}, \mathbf{m}_k) \\ \overleftarrow{\mathbf{l}_k} &= LSTM(\overleftarrow{\mathbf{l}_{k+1}}, \mathbf{m}_k) \\ \mathbf{l}_k &= \overrightarrow{\mathbf{l}_k} + \overleftarrow{\mathbf{l}_k} \end{aligned} \tag{3}$$

where $LSTM(\cdot)$, $\overrightarrow{\mathbf{l}_k}$, $\overleftarrow{\mathbf{l}_k}$ denote the LSTM network, the forward encoding of element $\mathbf{m}_k$ and the reverse encoding of element $\mathbf{m}_k$, respectively. The forward encoding $\overrightarrow{\mathbf{l}_k}$ of $\mathbf{m}_k$ is calculated from $\mathbf{m}_k$ and the forward encoding $\overrightarrow{\mathbf{l}_{k-1}}$ of the previous element $\mathbf{m}_{k-1}$.

Finally, the vector sequence $L = (\mathbf{l}_1, \mathbf{l}_2 \ldots, \mathbf{l}_n)$ is obtained. According to the calculation of the BiLSTM, it is known that $\mathbf{l}_k$ considers the semantic information in the forward and backward directions. This allows the boundary feature $\mathbf{g}_k$ to be fully utilized.

Two-Way Fusion. The $\mathbf{l}_k$ output by BiLSTM cannot avoid losing part of the semantic information of $\mathbf{x}_k$. To solve this problem, $\mathbf{x}_k$ and $\mathbf{l}_k$ are fused. The calculation is as follows:

$$\mathbf{h}_k = LN(\mathbf{W}_e(LeakyReLU(\mathbf{W}_l[\mathbf{x}_k, \mathbf{l}_k]))) \tag{4}$$

where $\mathbf{W}_l \in \mathbb{R}^{d\times 2d}$ and $\mathbf{W}_e \in \mathbb{R}^{d\times d}$ are trainable matrices, $[\cdot,\cdot]$ denotes concatenation, $LeakyReLU(\cdot)$ denotes the activation function LeakyReLU, and $LN(\cdot)$ denotes the LayerNorm layer. Thus, the final semantic representation vector sequence $H = (\mathbf{h}_1, \mathbf{h}_2 \ldots, \mathbf{h}_n)$ of the problem text sequence are obtained.

3.3 Decoder

A tree decoder [1] is adopted to generate expression tree. The first vector $\mathbf{h}_1$ (encoding of token CLS) of the coding sequence H output by CWFE is used to initialize the root node when decoding. Recursively, the decoder generates the embedding of each node and predicts the probabilities of number and operator candidates.

3.4 Model Training

Label Loss. Given the training data set $D = (P_i, T_i), 1 \leq i \leq N$, where N denotes the number of samples and T_i is the expression tree label of problem P_i. The label loss for each sample is calculated by following equation:

$$L_i = \sum_{t=1}^{m} -log\mathcal{P}(y_t|P_i) \tag{5}$$

where m is the length of T_i and y_t is the tth node of the prior traversal order of T_i.

Negative Label Definition and Negative Label Tree Generation. Suppose the label of node A generated by the tree decoder is -, ÷, or ∧; B and C are child nodes of A. We define that B's label is C's negative label and C's label is B's negative label. As shown in Fig. 2, the label of node1 is -; node2 and node3 are child nodes of node1. The negative label of node 2 and node 3 is ÷ and 1650, respectively.

By swapping left and right child nodes of operators -, ÷, and ∧ in the expression tree label, the negative label tree T^N is constructed, as shown in the lower right corner of Fig. 2.

Negative Label Loss. Negative label loss L_i^N reduces the probability that the predicted result of a node is a negative label, as shown by the orange arrow pointing downwards on the right side of Fig. 2. L_i^N is calculated from the following equation:

$$L_i^N = \sum_{t=1}^{m} \begin{cases} -log(1 - \mathcal{P}(y_t^N|P_i)) & if\ y_t^N \neq y_t \\ 0 & if\ y_t^N = y_t \end{cases} \tag{6}$$

where y_t^N is tth node of prior traversal order of negative label tree T^N; $y_t^N \neq y_t$ means the node has negative label.

Total Loss. The training objective is to minimize the total loss L_{total} of the model on the training set, which is calculated by the following equation:

$$L_{total} = \sum_{i=1}^{N} L_i + L_i^N \tag{7}$$

4 Experiment

4.1 Dataset, Baselines and Evaluation Metric

Dataset. CWFE-GTS-N is designed for Chinese MWPs solving, therefore two Chinese datasets are chosen for the experiment.

Math23K: The dataset Math23K [9] is the most commonly used dataset in the field of MWPs, containing 22,161 training samples and 1,000 test samples.

Ape-clean$^{\sim}$: The dataset Ape-clean [14] is the cleaned version of Ape210K [15]. Since there are still some problems with Ape-clean, we have made some following changes.

- Remove samples that cannot convert math expression label to the correct expression tree label.
- Replace the operator ** with $\wedge$ because ** will affect the math expression label to convert into a correct expression tree label.
- Replace ((())/(())) in the problem text with the Chinese word “多少” because they are equivalent in meaning and the latter provides the correct word boundary information.

The modified Ape-clean is represented as Ape-clean$^{\sim}$. Ape-clean$^{\sim}$ contains 101,286 training samples and 2,393 test samples.

Baselines. The following methods are compared on the datasets Math23K and Ape-clean$^{\sim}$.

- **Math-EN** [10]: An integrated model combining RNN, CNN, Transformer, and designed expression regularization to reduce the output space.
- **GROUP-ATT** [17]: The Seq2Seq model with the group attention mechanism to extract global features, quantity-related features, quantity-pair features, and question-related features in MWPs respectively.
- **GTS** [1]: The tree decoder is designed by borrowing from the way humans think when solving MWPs, and its encoder is a bi-directional GRU network.
- **HGEN** [3]: A hierarchical heterogeneous graph encoder to extract more features in the text.
- **RPKHS** [4]: A hierarchical inference encoder for integrating word-level and sentence-level inference.
- **E-GTS**: GTS but with ELECTRA as encoder.

Evaluation Metric. In the field of MWPs, models are equipped with the ability to generate correct new expressions that are different from the expression label. For example, the generated expression 5+3 is different from the expression label 3+5, but their calculated values are equal. In this case, the generated expression is considered to be correct. Therefore, the final computation result is used as the evaluation metric in this paper.

4.2 Implementation Details

CWFE-GTS-N proposed in this paper is implemented on the Ubuntu system using PyTorch and trained on RTX3090. A pre-trained version of ELECTRA on a large 180G Chinese corpus [16] is used. The dimensionality of all hidden states of the encoder is set to 768. CWFE-GTS-N is trained for 80 epochs using

Table 2. Accuracy of different models.

	Math23K	Ape-clean$^{\sim}$
Math-EN [10]	66.7	73.8
GROUP-ATT [17]	69.5	75.7
GTS [1]	75.6	80.3
HGEN [3]	78.7	–
RPKHS [4]	83.9	–
E-GTS	84.3	85.6
CWFE-GTS-N(ours)	**85.9**	**86.4**

the AdamW optimization algorithm, and the mini-batch size is set to 64. The initial learning rates of the encoder and decoder are 5e-5 and 1e-3, respectively. The learning rate is halved when the decrease in loss is less than 0.1 times the current loss. The dropout is set to 0.5 to prevent overfitting of model and use a 5 beam search to obtain better generation results.

4.3 Experimental Results

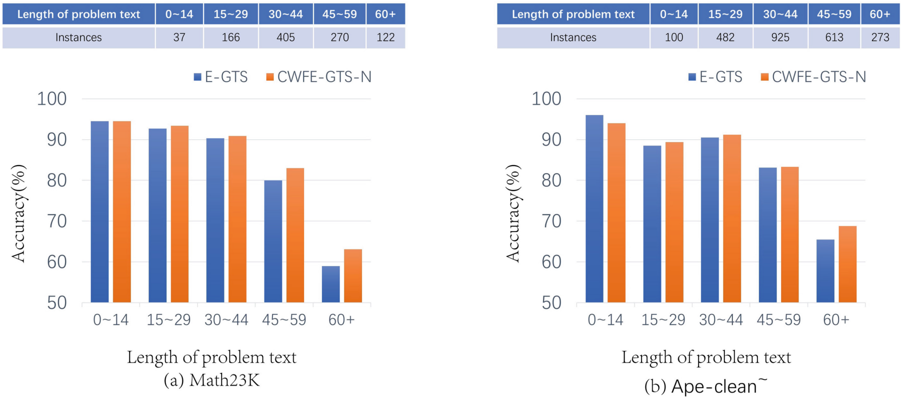
The symbol "-" denotes the code is not public. According to Table 2, there are several observations. First, the performance of GTS using the tree decoder exceeds the Seq2Seq class models (Math-EN, GROUP-ATT), which indicates GTS's tree decoder works well. Second, the models of HGEN, RPKHS, E-GTS, and CWFE-GTS-N use the same decoder, but the encoders are different resulting in large performance differences. Third, our CWFE-GTS-N outperforms the baseline models of HGEN, RPKHS, and E-GTS by 7.2%, 2%, and 1.6%, respectively, on dataset Math23K. CWFE-GTS-N improves 0.8% over E-GTS on dataset Ape-clean$^{\sim}$.

Ablation Study. CWFE-GTS-N uses CWFE as an encoder and adds negative label loss during training. To investigate the effect of CWFE and negative label loss on CWFE-GTS-N performance, an ablation experiment on Math23K is conducted. The results are shown in Table 3. Specifically, "w/o CWFE" means

Table 3. Accuracy of various configurations.

	Math23K
CWFE-GTS-N	85.9
w/o CWFE	85.3
w/o Neg-lab-loss	85.1
w/o (CWFE & Neg-lab-loss)	84.3

CWFE-GTS-N does not use CWFE as an encoder, but ELECTRA. “w/o Neg-lab-loss” indicates the CWFE-GTS-N removes the negative label loss during training. It can be seen that both CWFE and negative label loss have positive effects on the model.



Length of problem text	0~14	15~29	30~44	45~59	60+
Instances	37	166	405	270	122

Length of problem text	0~14	15~29	30~44	45~59	60+
Instances	100	482	925	613	273

(a) Math23K

(b) Ape-clean~

Fig. 3. Accuracy over different problem text lengths.

Impact of Problem Text Length on Model Performance. The long problem text is a challenge for the encoder’s coding capability. In general, the longer text means that the problem is more difficult. Therefore, it is necessary to investigate the impact of the text lengths on the result accuracy. The number of test instances over different text lengths is recorded, and the accuracies of corresponding test instances are compared with the E-GTS in Fig. 3. There are several observations. First, generally, the performance of each model degrades with the increasing length of text, which is reasonable due to increasing difficulty. Second, the difference between the performance of the two models becomes more obvious as the text length increases. The reason for this situation is that the probability of E-GTS neglecting keywords increases when the problem text is longer. In contrast, CWFE can fuse the character and word boundary information to provide a more accurate semantic representation for the decoder. Therefore CWFE-GTS-N is more advantageous in solving more complex with long problem text MWPs.

Case Study. To better understand the effectiveness of CWFE-GTS-N, four cases are provided in Table 4. Our analyses are summarized as follows:

- Case 1 is the example in the Introduction. CWFE-GTS-N uses character and word feature information to correctly determine the question is asking about the surface area of a “正方体 (cube)” rather than the surface area of a “正方 (square)”.

Table 4. Typical cases.

Case 1：一个棱长为6分米的正方体木块的表面积=多少平方分米.(How many square decimetres is the surface area of a cube wood block with a length of 6 decimetres?) E-GTS：$6 * 6(\times)$ CWFE-GTS-N：$6 * 6 * 6(\surd)$
Case 2：4（1）班50人中有28人喜欢吃香蕉,27人喜欢吃苹果,14人两种水果都喜欢吃．请问有多少人喜欢的不是这两种水果？(In class 4(1), there are 28 people who like bananas, 27 people who like apples, and 14 people who like both kinds of fruits. How many of them like other than these two fruits?) E-GTS：$28 + 27 - 14(\times)$ CWFE-GTS-N：$50 - [(27 + 28) - 14](\surd)$
Case 3：鸭比鸡多210只，鸡的只数和鸭的只数比是2：5．鸭和鸡共有多少只？(There are 210 more ducks than chickens, and the ratio of the number of chickens to the number of ducks is 2:5. How many ducks and chickens are there?) E-GTS：$210 \div (2 - 5) * (5 + 2)(\times)$ CWFE-GTS-N：$210 \div (5 - 2) * (2 + 5)(\surd)$
Case 4：粮食仓库存有大米1650袋，正好是存有的面粉袋数的6倍．粮库中存有的面粉比大米少多少袋？(The grain store has 1650 bags of rice, which is exactly 6 times the number of bags of flour in the store. How many fewer bags of flour than rice are there in the grain store?) E-GTS：$1650 \div 6 - 1650(\times)$ CWFE-GTS-N：$1650 - 1650 \div 6(\surd)$

- Case 2 is similar to Case 1. E-GTS does not consider “不” and “是” as a whole of “不是 (not)”, therefore mistakenly considers that the question asks how many people like both fruits. CWFE-GTS-N avoids this error.
- Case 3 and Case 4 illustrate the effectiveness of negative label loss. It makes the model correctly resolve the position of the operator’s operands.

5 Conclusion

In this paper, CWFE and negative label loss are proposed. CWFE provides the decoder with both character and word feature information in the text of Chinese MWPs. Word feature information is important for solving Chinese MWPs, which can reduce the probability of the model neglecting keywords. In addition, to prevent the model from incorrectly resolving the positions of operands, such as divisor and dividend, we introduce negative label loss in training. The proposed model CWFE-GTS-N uses CWFE as encoder and adds negative label loss function during training. Experiments show that CWFE-GTS-N outperforms several baseline models of HGEN, RPKHS, and E-GTS. In the future, we will explore how to design a suitable pre-training task for CWFE to mine more useful semantic information in the text of MWPs.

Acknowledgements. This paper is supported by the National Natural Science Foundation of China No.62177015.

References

1. Xie, Z., Sun, S.: A goal-driven tree-structured neural model for math word problems. In: IJCAI, pp. 5299–5305 (2019)
2. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Workshop of NIPS (2014)
3. Zhang, Y., et al.: Hgen: learning hierarchical heterogeneous graph encoding for math word problem solving. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 816–828 (2022)
4. Yu, W.J., et al.: Improving math word problems with pre-trained knowledge and hierarchical reasoning. In: EMNLP, pp. 3384–3394 (2021)
5. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: pre-training text encoders as discriminators rather than generators. In: ICLR (2020)
6. Fletcher, C.R.: Understanding and solving arithmetic word problems: a computer simulation. Behav. Res. Methods Instruments Comput. **17**(5), 565–571 (1985)
7. Bakman, Y.: Robust understanding of word problems with extraneous information. arXiv preprint math/0701393 (2007)
8. Kushman, N., Artzi, Y., Zettlemoyer, L., Barzilay, R.: Learning to automatically solve algebra word problems. In: ACL, pp. 271–281 (2014)
9. Wang, Y., Liu, X., Shi, S.: Deep neural solver for math word problems. In: EMNLP, pp. 845–854 (2017)
10. Wang, L., Wang, Y., Cai, D., Zhang, D., Liu, X.: Translating a math word problem to a expression tree. In: EMNLP, pp. 1064–1069 (2018)
11. Zhang, J., Lee, R.K.W., Lim, E.P., Qin, W., Wang, L., Shao, J., Sun, Q.: Teacher-student networks with multiple decoders for solving math word problem. In: IJCAI (2020)
12. Li, Z., et al.: Seeking patterns, not just memorizing procedures: contrastive learning for solving math word problems. In: Findings of ACL, pp. 2486–2496 (2022)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Liang, Z., et al.: Mwp-bert: numeracy-augmented pre-training for math word problem solving. In: Findings of NAACL, pp. 997–1009 (2022)
15. Zhao, W., et al.: Ape210k: a large-scale and template-rich dataset of math word problems. arXiv preprint arXiv:2009.11506 (2020)
16. Cui, Y., et al.: Revisiting pre-trained models for Chinese natural language processing. In: Findings of EMNLP, pp. 657–668 (2020)
17. Li, J., et al.: Modeling intra-relation in math word problems with different functional multi-head attentions. In: ACL, pp. 6162–6167 (2019)

MarkBERT: Marking Word Boundaries Improves Chinese BERT

Linyang Li[1], Yong Dai[2], Duyu Tang[2], Xipeng Qiu[1(✉)], Zelin Xu[3],
and Shuming Shi[2]

[1] Fudan University, Shanghai, China
{linyangli19,xpqiu}@fudan.edu.cn
[2] Tencent AI Lab, Shenzhen, China
{yongdai,duyutang,shumingshi}@tencent.com
[3] Peng Cheng Lab, Shenzhen, China

Abstract. We present a Chinese BERT model dubbed MarkBERT that uses word information in this work. Existing word-based BERT models regard words as basic units, however, due to the vocabulary limit of BERT, they only cover high-frequency words and fall back to character level when encountering out-of-vocabulary (OOV) words. Different from existing works, MarkBERT keeps the vocabulary being Chinese characters and inserts boundary markers between contiguous words. Such design enables the model to handle any words in the same way, no matter they are OOV words or not. Besides, our model has two additional benefits: first, it is convenient to add word-level learning objectives over markers, which is complementary to traditional character and sentence-level pretraining tasks; second, it can easily incorporate richer semantics such as POS tags of words by replacing generic markers with POS tag-specific markers. With the simple markers insertion, MarkBERT can improve the performances of various downstream tasks including language understanding and sequence labeling. (All the codes and models will be made publicly available at https://github.com/).

Keywords: Pretrain Model · Chinese Segmentation

1 Introduction

Chinese words can be composed of multiple Chinese characters. For instance, the word 地球 (earth) is made up of two characters 地 (ground) and 球 (ball). However, there are no delimiters (i.e., space) between words in written Chinese sentences. Traditionally, word segmentation is an important first step for Chinese natural language processing tasks [1]. Instead, with the rise of pretrained models [6], Chinese BERT models are dominated by character-based ones [3,4,16–18], where a sentence is represented as a sequence of characters. There are several attempts at building Chinese BERT models where word information is considered. Existing studies tokenize a word as a basic unit [15], as multiple characters

Work during internship in Teceng AI Lab.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 325–336, 2023.
https://doi.org/10.1007/978-3-031-44693-1_26

[4] or a combination of both [7,10,22]. However, due to the limit of the vocabulary size of BERT, these models only learn for a limited number (e.g., 40K) of words with high frequency. Rare words below the frequency threshold will be tokenized as separate characters so that the word information is neglected.

In this work, we present a simple framework, MarkBERT, that considers Chinese word information. Instead of regarding words as basic units, we use character-level tokenizations and inject word information via inserting special markers between contiguous words. The occurrence of a marker gives the model a hint that its previous character is the end of a word and the following character is the beginning of another word. Such a simple model design has the following advantages. First, it avoids the problem of OOV words since it deals with common words and rare words (even the words never seen in the pretraining data) in the same way. Second, the introduction of marker allows us to design word-level pretraining tasks (such as replaced word detection illustrated in Sect. 3), which are complementary to traditional character-level pretraining tasks like masked language modeling and sentence-level pretraining tasks like next sentence prediction.

In the pretraining stage, we force the markers to understand the contexts around them while serving as separators between words. We train our model with two pretraining tasks. The first task is masked language modeling and we also mask markers such that word boundary knowledge can be learned since the pre-trained model needs to recognize the word boundaries within the context. The second task is replaced word detection. We replace a word with artificially generated words and ask the markers behind the word to predict whether the word is replace. Such a process will force the markers to serve as discriminators therefore can learn more word-boundary information within the context. With these two pretraining tasks, we train the MarkBERT model initialized from BERT-Chinese models and obtain considerable improvements.

We conduct extensive experiments on various downstreams tasks including named entity recognition tasks (NER) and natural language understanding tasks. On the NER task, we demonstrate that MarkBERT can significantly surpass baseline methods on both MSRA and OntoNotes datasets [8,23]. Compared with other word-level Chinese BERT models, we conduct experiments and observe that MarkBERT performs better on text classification, keyword recognition, and semantic similarity tasks in the CLUE benchmark datasets.

2 Related Work

We describe related work on injecting word information to Chinese BERT and the use of marker in natural language understanding tasks.

2.1 Chinese BERT

Pre-trained models exemplified by BERT [6] and RoBERTa [4] have been proved successful in various Chinese NLP tasks [5,20]. Existing Chinese BERT models that incorporate word information can be divided into two categories. The first category uses word information in the pretraining stage but represents a text as a sequence of characters when the pretrained model is applied to downstream tasks. For example, Roberta-wwm [4] use the whole-word-masking strategy that masks word spans and predicts continuously multiple masked positions. Lattice-BERT [10] incorporate lexicon information by concatenating the lexicons along with character-level context. The second category uses word information when the pretrained model is used in downstream tasks. For example, WoBERT [15] uses a word-level vocabulary instead of characters. If a word 地球 is included in the vocabulary, its constitutes 地 and 球 will not be considered as input tokens. AMBERT [22] go one step further by constructing two independent encoders that encode character-level and word-level information separately and concatenate them at the top layers of two encoders. Similarly, LICHEE [7] encode both character-level and word-level information. They move the information aggregation stage to the embedding level.

2.2 Marker Insertion in NLU

The idea of inserting markers is explored in entity-related natural language understanding tasks, especially in relation classification. Given a subject entity and an object entity as the input, existing work inject untyped markers [14,17] or entity-specific markers [24] around the entities, and make better predictions of the relations of the entities.

3 MarkBERT Pre-training

In this section, we first introduce the background of character level Chinese pre-trained models; then we introduce the structure of our MarkBERT model. After describing the structure of MarkBERT, we introduce the training process of the MarkBERT. Finally, we provide details of the entire training process.

3.1 MarkBERT Model

To make better use of word-level information in Chinese pre-training, we introduce a simple framework called MarkBERT. We insert markers between word spans to give explicit boundary information for the model pre-training.

As seen in Fig. 1, we first use a segmentation tool to obtain word segmentations, then we insert special markers between word spans as separators between characters. These markers are treated as normal characters so they take positions in the transformers structure. Plus, they can also be masked for the mask language modeling task to predict, therefore the encoding process needs to be aware of predicting word boundaries rather than simply filling in masks from the context. The mask prediction task becomes more challenging since predicting the masks correctly requires a better understanding of the word boundaries. In this way, the model is still character-level encoded while it is aware of word boundaries since word-level information is given explicitly.

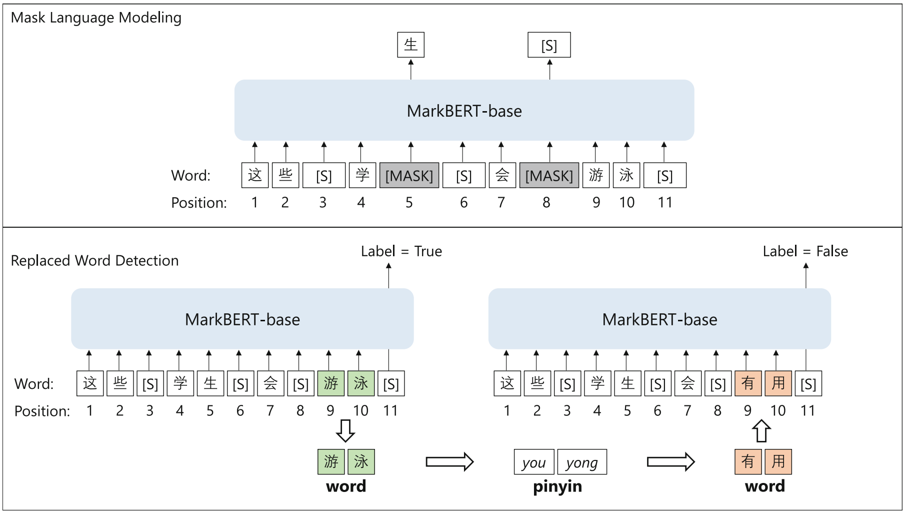


Fig. 1. Illustration of the predicting tasks of Masked Language Modeling and Replaced Word Detection. Here, `[S]` is the inserted markers.

3.2 Replaced Word Detection

Inserting special markers allows the pre-trained model to recognize word boundaries while maintaining a character-level model. Further, these special markers can be used to construct a word-level pre-training task which can be complementary to the character-level masked language modeling task.

We construct a replaced word detection task as an auxiliary task to the masked language modeling task. We construct a bipolar classification task that detects whether the word span is replaced by a confusion word. Specifically, given a word span, we take the representations of the marker after it and make binary prediction.

When a word span is replaced by a confusion word, as seen in Fig. 1, the marker is supposed to make a "replaced" prediction labeled as "False". When the word spans are not changed, the marker will make an "unchanged" prediction labeled as "True". Therefore, suppose the representation of the i^{th} marker is x^i with label y^{true} and y^{false}, the replaced word detection loss is: $\mathcal{L} = -\sum_i [y \cdot log(x^i)]$. We add this loss term to the masked language modeling loss as a multi-task training process.

The construction of the confusion could be various. We adopt two simple strategies: (1) we use synonyms as confusions; (2) we use words that are similar in phonetics (pinyin) in Chinese. To obtain the synonyms, we use an external word embedding [23]. We calculate the cosine similarity between words and use the most similar ones as the synonyms confusions. To obtain the phonetic-based confusions, as seen in Fig. 1, we use an external tool to get the phonetics of the word and select a word that share the same phonetics as its confusions.

In this way, the markers can be more sensitive to the word span in the context since these markers are assigned to discriminate the representation type of the word spans before them. This process is similar to an ELECTRA [2] framework. MarkBERT uses the inserted markers to run the discrimination process inside the encoder and use external confusions instead of using another generator to build texts for the discriminator.

3.3 Pre-training

The pre-training process is a multi task framework consisting of mask language modeling task and replaced word detection task.

In the masked language modeling task, we employ both the masked language modeling strategy and the whole-word-masking strategy. In the replaced word detection task, as seen in Fig. 1, when the word span is replaced by confusion words, the model is supposed to correct the confusions. This correction process is similar to MacBERT [3]. For the confusion generation, we use synonyms and pinyin-based confusions. The synonyms are obtained by a synonym dictionary based on calculating the cosine similarity between the Chinese word-embeddings [23].

In our MarkBERT pre-training, the mask ratio is still 15% of the total characters. For 30% of the time, we do not insert any markers so that the model can also be used in a no-marker setting which is the vanilla BERT-style model. For 50% of the time we run a whole-word-mask prediction and for the rest we run a traditional masked language model prediction. In the marker insertion, for 30% of the time, we replace the word span with a phonetic(pinyin)-based confusion or a synonym-based confusion word and the marker will predict a phonetic(pinyin)-confusion marker or a synonym-confusion marker; for the rest of the time, the marker will predict a normal-word marker.

Therefore, we only calculate 15% percent of loss on these normal markers to avoid imbalance labels of the marker learning process. During fine-tuning on downstream tasks, we use the markers in the input texts. Also, we can save the markers and downgrade the model to a vanilla BERT-style model for easier usage.

3.4 Implementation Details in Pre-training

Pre-training Dataset Usage. We use a collection of raw Chinese texts containing Chinese wikipedia, Chinese novels, news. The entire data size is around 80B characters. We use a simple word segmentation tool Texsmart [21] to tokenize the raw data and obtain pos-tags. We use the same data preprocess framework used in BERT [6] which constructs documents containing multiple sentences with the length of the maximum token limit and randomly pick another document to train the next sentence prediction task.

Pre-training Settings. We initialize our model from the Roberta whole-word-mask model checkpoint [4]. Therefore, we use the same character-level vocabulary in training our boundary-aware model. We use both whole-word-mask and normal character mask strategies in the language model training since we aim to learn inner connections between characters in the given word which cannot be achieved by whole-word-masking alone.

We train the model with a maximum sequence length of 512 for the entire training time. With the markers inserted, the actual maximum sequence length is smaller but we maintain the length as 512 to keep coordinated with previous pre-trained models. We use the ADAM optimizer [9] used in BERT with a batch size 8,192 on 64x Tesla V100 GPUs. We set the learning rate to 1e-4 with a linear warmup scheduler. We run the warmup process for 10k steps and train 100k steps in total.

4 Experiments

4.1 NER Task

In the NER task, we use the MSRA [11] and Ontonotes [19] datasets with the same data-split [12,13].

We establish several strong baselines to explore the effectiveness of our MarkBERT. In language understanding tasks, we compare with the RoBERTa-wwm-ext [4] baseline, which is a whole-word-mask trained Chinese pre-trained models. We also further pre-train the RoBERTa model denoted as RoBERTa (ours) and the WoBERT model denoted as WoBERT (ours) based on our collected data which is the same data used in pre-training MarkBERT to make fair comparisons with our model. In the NER task, we compare with FLAT-BERT [12] and Soft-Lexicon [13] which are state-of-the-art models on the NER task which incorporate lexicons in the transformers/LSTM structure.

4.2 Language Understanding Task

We also conduct experiments on language understanding tasks. We use various types of tasks from the CLUE benchmark [20]. We use classification tasks such as TNEWS, IFLYTEK; semantic similarity task (AFQMC); coreference resolution task (WSC); keyword recognition (CSL); natural language inference task (OCNLI).

Besides the BERT-style baselines used in the NER task, we also use the word-level information enhanced models as baselines to make comparisons in the language understanding tasks. We use:

- WoBERT [15]: a word-level Chinese pre-trained model initialized from the BERT BASE pre-trained weights. It has a 60k expanded vocabulary containing commonly used Chinese words.

- AMBERT [22]: a multi-granularity Chinese pre-trained model with two separated encoders for words and characters. The encoding representation is the character-level representation concatenated by the word-level representation;
- LICHEE [7]: a multi-granularity Chinese pre-trained model that incorporates word and character representations at the embedding level.
- Lattice-BERT [10]: the state-of-the-art multi-granularity model that uses lexicons as word-level knowledge concatenated to the original input context.

4.3 Downstream Task Implementations

We use the FastNLP toolkit[1] to implement the NER experiment; We use the Huggingface Transformers to implement all experiments.

For the NER task, we follow the implementation details given in the Transformers toolkit.[2] For the language understanding tasks, we follow the implementation details used in the CLUE benchmark official website and the fine-tuning hyper-parameters used in Lattice-BERT [10].

In the NER task, we use the marker-inserted inputs in the MarkBERT since we intend to incorporate the word boundary information in recognizing entities. We use the model with the best development performance to obtain the test set result. We make a thorough discussion on this topic in the later section. In the NER evaluation process, we label the inserted marker with the same label as its former token and follow the standard BMESO evaluation process [12,13].

In the NLU tasks, we use the CLUE benchmark datasets to test our model. For the TNEWS task, we run the raw classification results without using the keywords augmentation which is no longer a natural context. For the IFLYTEK task, we split the context and use the average of the split texts prediction since the average sequence exceeds the max sequence length. We leave the experiment results '-' if they are not listed in the official website.[3]

Table 1. NER results on the MSRA and OntoNotes dataset.

	MSRA (Test)			OntoNotes (Dev)			OntoNotes(Test)		
	Acc	Recall	F1	Acc	Recall	F1	Acc	Recall	F1
BERT [6]	94.9	94.1	94.5	74.8	81.8	78.2	78.0	75.7	80.3
RoBERTa [4]	95.3	94.9	95.1	76.8	80.7	78.7	77.6	83.5	80.5
FLAT-BERT [12]	–	–	96.1	–	–	–	–	–	81.8
Soft-Lexicon [13]	95.8	95.1	95.4	–	–	–	83.4	82.2	82.8
RoBERTa (ours)	95.7	94.8	95.2	80.3	76.4	78.3	78.8	83.4	81.1
MarkBERT (ours)	**96.1**	**96.0**	**96.1**	**81.2**	**81.4**	**81.3**	81.7	**83.7**	82.7

[1] https://github.com/fastnlp/fastNLP.

[2] https://github.com/huggingface/transformers.

[3] https://github.com/CLUEbenchmark/CLUE.

4.4 Results on NER Task

In Table 1, our proposed boundary-aware MarkBERT outperforms all baseline models including pre-trained models and lexicon-enhanced models.

Compared with the baseline methods, our proposed MarkBERT with markers inserted between words can lift performances by a large margin. We can observe that compared with the baseline method RoBERTa(ours) which uses word-level information by pretraining with the whole-word mask strategy, MarkBERT can significantly improve the performances in all datasets. When we insert markers using the same tokenization process used in pre-training MarkBERT in fine-tuning the MarkBERT in the NER task, we obtain a considerable performance improvement, indicating that the inserted markers catch some important fine-grained information that helps improve entity understanding. Further, when compared with previous state-of-the-art methods such as Soft-Lexicon [13] and FLAT [12] which use a combination of lexicon-enhanced LSTMs/transformers and BERT, our model can also achieve similar performance while we do not incorporate any lexicon information which is essential in Chinese language.

Therefore, we can conclude that MarkBERT can improve the NER task with a simple marker insertion strategy without complex lexicons therefore can be widely used in sequence labeling tasks.

Table 2. Evaluation results on the language understanding tasks.

	Datasets					
	TNEWS	IFLYTEK	AFQMC	OCNLI	WSC	CSL
DEVELOPMENT						
BERT [6]	56.09	60.37	74.10	74.70	79.22	81.02
RoBERTa [4]	57.51	60.80	73.80	75.01	82.20	81.22
RoBERTa (ours)	57.95	60.85	74.58	75.32	84.02	81.85
WoBERT (ours)	57.01	61.10	72.80	75.00	82.72	-
MarkBERT (ours)	**58.40**	60.68	**74.89**	**75.88**	**84.60**	-
TEST						
BERT [6]	56.58	60.29	73.70	-	62.00	80.36
RoBERTa [4]	56.94	60.31	74.04	-	67.80	81.00
AMBERT [22]	-	59.73	73.86	-	78.27	85.70
LICHEE [7]	-	60.94	73.65	-	81.03	84.51
BERT [10]	-	62.20	74.00	-	79.30	81.60
Lattice-BERT [10]	-	**62.90**	74.80	-	**82.40**	84.00
RoBERTa (ours)	57.42	61.00	73.63	72.67	79.86	81.83
MarkBERT (ours)	**58.05**	62.57	**74.87**	**73.06**	81.72	**85.73**

4.5 Results on Language Understanding

Table 2 shows that comparing with the RoBERTa model that uses the same pre-training data, MarkBERT is superior in all tasks. This indicates that the learned representations contain more useful information for the downstream task fine-tuning. The word-level model WoBERT (ours) trained with the same data used in MarkBERT only achieves a slightly higher accuracy in the IFLYTEK dataset which might because the IFLYTEK dataset contains very long texts where word-level model is superior since it can process more contexts while the total sequence lengths of character level and word level model are both 512.

When comparing with previous works that focus on word-level information, MarkBERT achieves higher performances than the multi-grained encoding method AMBERT as well as LICHEE which incorporates word information as an additional embedding. We can assume that adding word-level information through *horizontal* markers is more effective than *vertically* concatenating word-level information. When comparing with the LatticeBERT model, our method can still reach a competitive level of performance, meanwhile the relative improvements of our model is larger than the improvements of the LatticeBERT model. Please note that the lexicons used in LatticeBERT training actually contains more segmentation possibilities which can significantly increase the downstream task performance over the word segmentation based methods [23]. The basic idea of incorporating lexicons is parallel with the marker insertion framework. MarkBERT makes use of word-level information in a different perspective.

Table 3. Ablation Studies on the NER and the language understanding tasks using dev set results.

	Datasets				
	MSRA	Ontonotes	TNEWS	IFLYTEK	AFQMC
DEVELOPMENT	F1	F1	Acc	Acc	Acc.
MarkBERT	**96.1**	**82.7**	**58.4**	60.6	**74.8**
MarkBERT-rwd-pho	95.8	81.7	58.0	60.8	74.3
MarkBERT-rwd-syn	95.8	81.7	58.0	60.9	74.5
MarkBERT-MLM	95.8	81.3	58.0	60.7	74.6
MarkBERT-w/o marker	95.5	79.2	58.2	**61.0**	74.5
RoBERTa (ours)	95.1	78.2	57.9	60.8	74.5

4.6 Model Analysis

In this section, we conduct ablation experiments to explore the effectiveness of each parts in our MarkBERT framework in different tasks.

We test different variants of MarkBERT:

- MarkBERT-MLM only considers the MLM task without the replaced word detection task; the masked language model will predict masked tokens as well as inserted markers.
- MarkBERT-rwd is a version that removes phonetics words or synonyms separately in the replaced word detection process.
- MarkBERT-w/o marker is a version that removed markers which is the same as the vanilla BERT model.

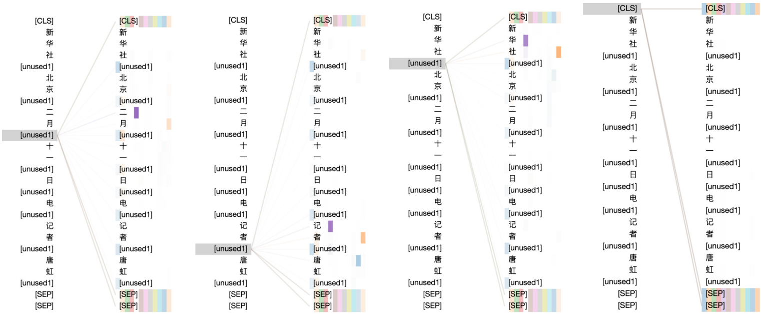
MarkBERT-MLM Without RWD. To explore which parts in MarkBERT is more effective, we conduct an experiment as seen in Table 3. We only use the masked language modeling task while inserting markers without using the replaced word detection task. The model only considers inserted markers and masked language modeling tasks, while the markers will be masked and predicted as well.

As seen, the MarkBERT -MLM model gains significant boost in the NER task, indicating that word boundary information is important in the fine-grained task.

In the CLUE benchmark, the situation becomes different: in the IFLYTEK task, inserting markers will hurt the model performance which is because the sequence length exceeds the maximum length of the pre-trained model. Therefore, inserting markers will results in a lost of contexts. Generally, inserting markers is important in downstream task fine-tuning. The explicit word boundary information helps MarkBERT learn better contextualized representations.

Replaced Word Detection. We also test the effectiveness of the additional replaced word detection task. Specifically, we separate two confusion strategies and use phonetics and synonyms confusions solely.

As seen in Table 3, when the marker learning only includes phonetic (pinyin) confusions, the performances in the fine-tuning tasks are similar with the MarkBERT -MLM model, indicating that the phonetic confusions have a slight improvement based on the inserted markers. When the word spans are replaced by synonyms only, the performances are slightly lower than using both phonetic and synonym confusions, indicating that augmentation using various types of confusions is helpful.


[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]
[CLS] 新 华 社 [unused1] 北 京 [unused1] 二 月 [unused1] 十 一 [unused1] 日 [unused1] 电 [unused1] 记 者 [unused1] 唐 虹 [unused1] [SEP] [SEP]

(a) (b) (c) (d)

Fig. 2. Visualization of attention of the markers selected from a random layer. We use `[unused1]` in the BERT vocabulary as the inserted marker.

MarkBERT-w/o Marker. Inserting markers is the key idea of solving the character and word dilemma in Chinese encoding. In the NER task, inserting markers is important, indicating that MarkBERT structure is effective in learning word boundaries for tasks that requires such fine-grained representations. In the NLU tasks, without inserting markers, MarkBERT-w/o marker can still achieve similar performances with the baseline methods, indicating that MarkBERT can also be used as a vanilla BERT model for easy usage in language understanding tasks.

Visualization of Marker Attentions. To further explore how the markers work in the encoding process, we use the attention visualization tool to show the attention weights of the inserted markers. We explore the attention weights on the pre-trained MarkBERT and the fine-tuned model based on the Ontonotes NER task. As seen in Fig. 2, in some heads of the representations of the inserted markers, the attentions focus on the local semantics (e.g. in Fig. 2(a), the marker is attended to '二' (second) and '月'(month) in the head colored with purple and orange, indicating that the marker learn the context of the word '二月' (February). Further, the special tokens are the mostly focused as seen in Fig. 2(d).

5 Conclusion and Future Work

In this paper, we have introduced MarkBERT, a simple framework for Chinese language model pre-training. We insert special markers between word spans in the character-level encodings in pre-training and fine-tuning to make use of word-level information in Chinese. We test our proposed model on the NER tasks as well as natural language understanding tasks. Experiments show that MarkBERT makes significant improvements over baseline models.

References

1. Chang, P.C., Galley, M., Manning, C.D.: Optimizing Chinese word segmentation for machine translation performance. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 224–232 (2008)
2. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
3. Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G.: Revisiting pre-trained models for Chinese natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 657–668. Association for Computational Linguistics, Online (Nov 2020), https://www.aclweb.org/anthology/2020.findings-emnlp.58
4. Cui, Y., et al.: Pre-training with whole word masking for Chinese bert. arXiv preprint arXiv:1906.08101 (2019)

5. Cui, Y., et al.: A span-extraction dataset for Chinese machine reading comprehension. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5886–5891. Association for Computational Linguistics, Hong Kong, China, November 2019. https://doi.org/10.18653/v1/D19-1600. https://www.aclweb.org/anthology/D19-1600
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805
7. Guo, W., et al.: Lichee: improving language model pre-training with multi-grained tokenization. In: FINDINGS (2021)
8. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Lai, Y., Liu, Y., Feng, Y., Huang, S., Zhao, D.: Lattice-bert: leveraging multi-granularity representations in chinese pre-trained language models. arXiv preprint arXiv:2104.07204 (2021)
11. Levow, G.A.: The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pp. 108–117. Association for Computational Linguistics, Sydney, Australia, July 2006. https://aclanthology.org/W06-0115
12. Li, X., Yan, H., Qiu, X., Huang, X.: Flat: Chinese ner using flat-lattice transformer. arXiv preprint arXiv:2004.11795 (2020)
13. Ma, R., Peng, M., Zhang, Q., Huang, X.: Simplify the usage of lexicon in Chinese ner. arXiv preprint arXiv:1908.05969 (2019)
14. Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: distributional similarity for relation learning. arXiv preprint arXiv:1906.03158 (2019)
15. Su, J.: Wobert: Word-based chinese bert model - zhuiyiai. Technical report (2020). https://github.com/ZhuiyiTechnology/WoBERT
16. Sun, Y., et al.: Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation (2021)
17. Sun, Y., et al.: Ernie: enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
18. Sun, Z., et al.: Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. arXiv preprint arXiv:2106.16038 (2021)
19. Weischedel, R., et al.: Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium, Philadelphia, PA 23 (2013)
20. Xu, L., et al.: Clue: a Chinese language understanding evaluation benchmark. arXiv preprint arXiv:2004.05986 (2020)
21. Zhang, H., et al.: Texsmart: a text understanding system for fine-grained ner and enhanced semantic analysis. arXiv preprint arXiv:2012.15639 (2020)
22. Zhang, X., Li, H.: Ambert: a pre-trained language model with multi-grained tokenization. arXiv preprint arXiv:2008.11869 (2020)
23. Zhang, Y., Yang, J.: Chinese ner using lattice lstm. arXiv preprint arXiv:1805.02023 (2018)
24. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. arXiv preprint arXiv:2010.12812 (2020)

MCVIE: An Effective Batch-Mode Active Learning for Multi-label Text Classification

Xuan Cheng[1,2], Feng Zhou[1,2], Qing Wang[3](✉), Yitong Wang[3], and Yiting Wang[3]

[1] Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing, China
{chengxuan,zfeng}@bupt.edu.cn
[2] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
[3] JiuTian Team, China Mobile Research Institute, Beijing, China
{wangqingai,wangyitongyjy,wangyiting}@chinamobile.com

Abstract. Data labeling for multi-label text is a challenging task in natural language processing, and active learning has emerged as a promising approach to reduce annotation effort while improving model performance. The primary challenge in multi-label active learning is to develop query strategies that can effectively select the most valuable unlabeled instances for annotation. Batch-mode active learning approaches, which select a batch of informative and diverse instances in each iteration, have been considered useful for improving annotation efficiency. However, challenges such as incomplete information ranking and high computational costs still hinder the progress of batch-mode methods. In this paper, we propose MCVIE, a novel batch-mode active learning method for multi-label text. MCVIE employs a two-stage active learning query strategy. Firstly, we combine two measures of prediction uncertainty and category vector inconsistency to calculate the basic information score for each example-label pair. Then, we use the Euclidean distance of text feature vectors to iteratively select diverse and informative example-label pairs for annotation. Experimental results on three benchmark datasets demonstrate that MCVIE outperforms other competitive methods.

Keywords: Multi label · Active learning · Information score

1 Introduction

Multi-label text classification, a variation of multi-class classification, has drawn considerable attention due to its applications in various fields such as sentiment analysis [6], spam detection [9], and news categorization [7]. However, label acquisition for multi-label text is difficult and expensive; even some labels require annotators with specialized knowledge [18]. And the serious imbalance

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 337–348, 2023.
https://doi.org/10.1007/978-3-031-44693-1_27

for each category in the multi-label text datasets further increased the difficulty of annotation tasks.

Multi-label active learning (MLAL) has been proven to be an effective approach to addressing the above issues [1,4,10,11]. MLAL is primarily focused on how to select the most valuable unlabeled instances for labeling to learn a better classifier. According to the number of instances queried in each iteration, existing mainstream MLAL methods can be divided into myopic strategies and batch-mode strategies. Myopic strategies [1,10] select only one unlabeled instance with a model update in each iteration, which can prevent the selection of similar instances for labeling. However, a large number of iterations are required to accumulate sufficient labeled instances, resulting in tremendous time consumption. Batch-mode strategies [4,11] query a batch of unlabeled instances in each iteration by taking both uncertainty and diversity into consideration. Therefore, bath-mode strategies could provide a parallel annotation environment for multi-label text classification, significantly increasing efficiency.

To effectively select a batch of valuable unlabeled instances, example-label-based selection methods are more preferred than example-based selection methods. Example-based selection methods [10,16] select valuable instances with full class labels for annotation. But they may overlook cases where some class labels are not informative, particularly for the entire batch. Moreover, annotators have to scrutinize each label of the selected instance meticulously [4], increasing the time cost. Example-label-based selection methods [4,19] calculate candidate examples for each unlabeled class label, rank them at a finer granularity, and just select example-label pairs with the highest information score.

However, research into batch-mode example-label-based MLAL is limited. Most relevant MLAL strategies calculate an information score for candidates with uncertainty and diversity. But they fail to account for the label space of multi-label text data, leading to incomplete information ranking of unlabeled example-label pairs. Moreover, selecting a batch of data based solely on the information score may result in a large number of similar samples. Existing approaches may result in low labeling efficiency and fail to effectively improve the accuracy of the classifier. These issues render batch-mode example-label-based MLAL research inefficient for labeling multi-label text data.

In this paper, we propose MCVIE, a novel batch-mode example-label-based MLAL for multi-label text data. Specifically, 1) Based on the particularity of the label space of multi-label text data, we design a two-stage query strategy. MCVIE first calculates basic information score for unlabeled example-label pairs by combining measures of prediction uncertainty and category vector inconsistency. Next, MCVIE iteratively selects diverse and informative example-label pairs for annotation by employing Euclidean distance of feature vectors. 2) Considering the extreme imbalance of multi-label text data, we devise a variable threshold optimization evaluation using Micro and Macro F1 score. 3) Experimental results on three public multi-label text datasets demonstrate the effectiveness of MCVIE, compared to state-of-the-art methods. The code is available.[1]

[1] https://github.com/hanhan1214/active-learning-mcvie.

2 Related Work

Active Learning. Active learning (AL) is a type of machine learning in which query strategies interactively select the most valuable instances, thereby reducing the workload of human labeling. AL-related research has been going on for decades and is widely applied in many fields, including various image and text classifications [8,21]. Traditional AL methods rely on the predicted probability of the classifier to find the most uncertain instances [2]. However, unilaterally considering uncertainty is not sufficient; therefore, some novel approaches also take diversity [13] and representativeness [14] into consideration.

Multi-label Text Classification. In contrast to general text classification tasks, multi-label text classification involves automatically categorizing texts into the most relevant subset of the given label collection. Some traditional machine learning methods treat this task as multiple binary classification problems, training a binary classifier for each class independently [3]. With the emergence of deep learning, CNNs and attention mechanisms are widely used to tackle this challenging task [15,22]. Recently, pre-trained transformer-based models, such as BERT, have achieved state-of-the-art results in most NLP tasks, including multi-label text classification. Hence, we employ distilBERT[2], a small, fast, cheap, and light version made by distilling BERT base [12], as a suitable multi-label text classifier instead of traditional SVMs [5]. This allows us to better reflect the application scenario of AL and obtain more accurate and realistic evaluation metrics.

Multi-label Active Learning. MLAL is a subfield of AL that specifically focuses on multi-label learning. It incorporates a core query strategy that takes into account not only the uncertainty, diversity, and representativeness of the data but also the unique characteristics of multi-label data label space [4,10]. [20] evaluates candidate instances using the max loss strategy and the mean max loss strategy to query the most uncertain instances. [10] incorporates classifier predictions and inconsistency of predicted label sets to select the most informative instances for full labeling. [4] queries the example-label pairs with the highest informativeness and lowest redundancy by adopting a two-stage selection using uncertainty, label correlation, and label space sparsity. These previous studies infuse us with profound inspiration for our work, which will be described in detail in the next section.

3 Methodology

3.1 Problem Formulation

A multi-label text i can be expressed as $(\mathbf{X}_i, \mathbf{Y}_i)$, where $\mathbf{X}_i = \{x_{i1}, x_{i2}, ..., x_{id}\} \in R^d$ is the feature vector. $\mathbf{Y}_i = \{y_{i1}, y_{i2}, ..., y_{iq}\} \in \{0,1\}^q$ represents the category vector, where q is the number of class labels. More granularly, an example-label

[2] https://huggingface.co/distilbert-base-uncased.

pair can be denoted as $(\mathbf{X}_i, y_{ih})$, where y_{ih} denotes the h-th label of example i, $y_{ih} = 1$ indicates that i is a positive example for class h, while $y_{ih} = 0$ shows i is a negative instance for class h and does not belong to this category.

In a typical MLAL process, a fully labeled training set $\mathcal{L} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{l}$ acquired by initial random sampling and a fully unlabeled candidate data pool $\mathcal{U} = \{(\mathbf{X}_j, \mathbf{Y}_j)\}_{j=l+1}^{u+l}$ are preliminarily given, where $l \ll u$. The MLAL strategy iteratively selects the most valuable examples or example-label pairs from $\mathcal{U}$ for human annotation and moves the annotated data to $\mathcal{L}$, then a multi-label classifier Θ is retrained from scratch with updated $\mathcal{L}$. This process stops when the number of iterations or the accuracy of Θ reaches the advanced setting. The ultimate purpose of MLAL is to obtain a relatively accurate model without annotating entire unlabeled data, which enables us to achieve higher performance of Θ with far less annotation effort.

In this paper, MCVIE selects diverse and informative example-label pairs at each iteration through a two-stage query strategy we proposed. Firstly, we combine two measures of prediction uncertainty and category vector inconsistency to calculate the basic information score for each example-label pair. In the second stage, the basic score is further refined by incorporating text feature vectors to iteratively select example-label pairs. Figure 1 illustrates the main architecture of MCVIE, and details of the two-stage query strategy will be elaborated in the following sections.

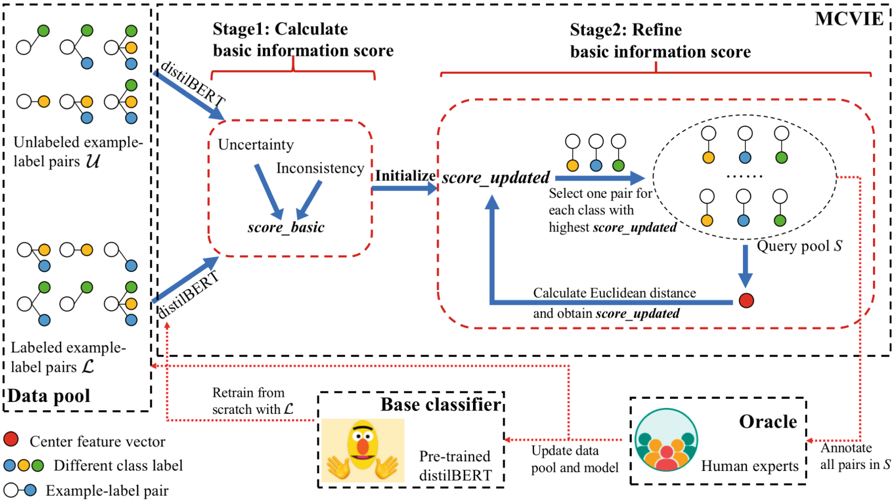

MCVIE
Stage1: Calculate basic information score
Stage2: Refine basic information score
distilBERT
distilBERT
Unlabeled example-label pairs $\mathcal{U}$
Labeled example-label pairs $\mathcal{L}$
Data pool
Uncertainty
Inconsistency
score_basic
Initialize
score_updated
Select one pair for each class with highest score_updated
......
Query pool S
Calculate Euclidean distance and obtain score_updated
Retrain from scratch with $\mathcal{L}$
Base classifier
Pre-trained distilBERT
Update data pool and model
Oracle
Human experts
Annotate all pairs in S
Center feature vector
Different class label
Example-label pair

Fig. 1. The main architecture of our proposed MCVIE.

3.2 Calculating Basic Information Score Stage

In the first stage, MCVIE calculates the basic information score for all candidate example-label pairs by combining measures of prediction uncertainty and category vector inconsistency. The following details the process.

At the beginning of each iteration, the classifier Θ is trained with the labeled set $\mathcal{L}$. Given a multi-label text i and a set of classes $C = (c_1, c_2, ..., c_q)$, Θ can output the predicted probability of i for each class label. The probability is expressed as a vector shaped like $\mathbf{P}_i = (p_{i1}, p_{i2}, ..., p_{iq}) \in [0,1]^q$.

Prediction Uncertainty Measure. Labeling example-label pairs with high uncertainty can provide more information for the model. For a certain class h, p_{ih} closed to 1 indicates the sample i is predicted as a positive example and closed to 0 is negative. Therefore, the closer the probability is to 0.5, the greater uncertainty of the sample. Based on these facts, the prediction uncertainty measure of an example-label pair $(\mathbf{X}_i, y_{ih})$ is computed as:

$$u(i,h) = \begin{cases} 2p_{ih} & \text{if } p_{ih} \leq 0.5 \\ 2(1-p_{ih}) & \text{if } p_{ih} > 0.5 \end{cases} \tag{1}$$

Category Vector Inconsistency Measure. In addition to considering the measure of prediction uncertainty, it is necessary to take into account the label space of multi-label data. Similar to diversity, texts have the same category vector, and the characteristics between them may be more similar. Therefore, the category vector of the example to be queried should be as far from the vector of the examples in $\mathcal{L}$ as possible, and this is referred to as category vector inconsistency.

To acquire category vector $\mathbf{Y}_i$ of example i, y_{ih} is the labeled value (i.e., the ground truth) if example-label pair $(\mathbf{X}_i, y_{ih})$ is already annotated. Otherwise, if it is not annotated, y_{ih} can be calculated using the probability p_{ih} and the threshold τ. Every p_{ih} greater than τ is considered as a positive prediction and smaller is negative, and the prediction y^*_{ih} for p_{ih} is computed as follows. More details about τ are in Sect. 4.3.

$$y^*_{ih} = \{c_h | p_{ih} > \tau\} \tag{2}$$

Given two category vectors $\mathbf{Y}_i$ and $\mathbf{Y}_j$ of text i and j, the contingency table is shown in Table 1. Element a is the count of h that $\mathbf{Y}_{ih} = \mathbf{Y}_{jh} = 1$, while b is the number of h that simultaneously $\mathbf{Y}_{ih} = 1$ and $\mathbf{Y}_{jh} = 0$. In the same way, the values of c and d can be calculated.

Table 1. Contingency table of two text category vectors.

$\mathbf{Y}_i/\mathbf{Y}_j$	1	0
1	a	b
0	c	d

The entropy distance is generally used to calculate the inconsistency between the structures of two binary vectors. The normalized entropy distance, expressed as d_E, between two category vectors is calculated as:

$$d_E\left(\mathbf{Y}_i, \mathbf{Y}_j\right) = -\frac{H\left(\mathbf{Y}_i\right) + H\left(\mathbf{Y}_j\right) - 2H\left(\mathbf{Y}_i, \mathbf{Y}_j\right)}{2H\left(\mathbf{Y}_i, \mathbf{Y}_j\right)} \tag{3}$$

where the entropy of a vector $\mathbf{Y}$ is computed as:

$$H\left(\mathbf{Y}\right) = H(r_0) = -r_0 log_2(r_0) - (1-r_0)log_2(1-r_0) \tag{4}$$

where the r_0 represents the proportion of elements 0 in the category vector $\mathbf{Y}$.

And the cross entropy is calculated as:

$$H\left(\mathbf{Y}_i, \mathbf{Y}_j\right) = H(\frac{b+c}{q}) + \frac{b+c}{q}H(\frac{b}{b+c}) + \frac{a+d}{q}H(\frac{a}{a+d}) \tag{5}$$

The distance d_E is highly adaptable to identify the structure differences between two category vectors. However, here is a problem scenario when $\mathbf{Y}_i$ and $\mathbf{Y}_j$ is completely opposite in every class. For example, $\mathbf{Y}_i = (1,0,1,1,0,1)$ and $\mathbf{Y}_j = (0,1,0,0,1,0)$. The value of $d_E\left(\mathbf{Y}_i, \mathbf{Y}_j\right)$ is 0 which indicates i and j have the same structure but $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are utterly different in fact. To address this problem, we define d_U and assign the maximum value of d_U when this problem scenario occurs. And d_U is accessed as:

$$d_U\left(\mathbf{Y}_i, \mathbf{Y}_j\right) = \begin{cases} 1 & \text{if } a+d=0 \\ d_E\left(\mathbf{Y}_i, \mathbf{Y}_j\right) & \text{if } a+d>0 \end{cases} \tag{6}$$

The introduction of d_H resolves the issue of d_E being unable to handle two completely opposite vectors. Based on d_U, the category vector inconsistency measure for a candidate example-label pair $(\mathbf{X}_i, y_{ih})$ is evaluated as:

$$v(i,h) = \frac{1}{|\mathcal{L}|}\sum_{j\in\mathcal{L}} d_U\left(\mathbf{Y}_i, \mathbf{Y}_j\right) \tag{7}$$

Basic Information Score. Considering the two measures previously defined, we preliminary calculate basic information score for candidate unlabeled example-label pairs. And the score of example-label pair $(\mathbf{X}_i, y_{ih})$ is computed as:

$$score\_basic(i,h) = u(i,h) * v(i,h) \tag{8}$$

3.3 Refining the Basic Information Score Stage

A higher value of $score\_basic(i,h)$ indicates that the example-label pair $(\mathbf{X}_i, y_{ih})$ contains more information. However, selecting a batch of top-ranked pairs solely based on the basic information score may result in the batch of highly similar texts. To address this issue, we designed a second stage that refines the basic information score by incorporating the diversity of text feature vectors and iteratively selects the example-label pairs.

As the second stage illustrated in Fig. 1, specifically, we maintain a query pool S in this stage. At the beginning of this stage, S is initialized by selecting only one example-label pair with the highest basic information score for each class. Firstly, we compute the center text feature vector of S. Next, we calculate the Euclidean distance between candidate examples and the center feature vector. Text that is farther away from the center vector is considered to have greater diversity. So we utilize the natural logarithm to scale it to the $[0, 1]$ range and combining the basic information score to obtain *score_updated*, which is computed as:

$$score\_updated(i,h) = \begin{cases} score\_basic(i,h) * exp\left(-\frac{1}{\|\mathbf{X}_i - \overline{\mathbf{X}}\|_2}\right) & \text{if } i \notin S \\ 0 & \text{if } i \in S \end{cases} \tag{9}$$

where $\overline{\mathbf{X}} = \frac{\sum_{k \in S} \mathbf{X}_k}{|S|}$ is the center feature vector of S. Pairs from the same example are not allowed to be simultaneously selected in one MCVIE iteration because of feature diversity, so we set all *score_updated* of example-label pairs that the examples are already in S to 0.

Similar to the initialization of S, we select only one example-label pair with the highest *score_updated* for each class and move these pairs from $\mathcal{U}$ to S. At the end of each iteration of the second stage, we check whether the size of S has reached the setting batch size. If it has, the second stage ends and sends all pairs in S to the oracle (i.e., human expert). If not, the second stage continues with a new iteration.

After the human labeling, the base classifier Θ is retrained from scratch with updated $\mathcal{L}$. MCVIE repeats the two stages mentioned above to select more diverse and informative example-label pairs for annotation. The pseudo code for proposed MCVIE is summarized in Algorithm 1.

Algorithm 1: MCVIE

```
Input  : $\mathcal{L} \rightarrow$ labeled training set
         $\mathcal{U} \rightarrow$ unlabeled data pool
         $\Theta \rightarrow$ base classifier distilBERT
Output: Fine-tuned $\Theta$ & labeled example-label pairs $\mathcal{L}$
Random sampling and fully labeled a small set $M$, $\mathcal{L} \leftarrow \mathcal{L} \cup M$, $\mathcal{U} \leftarrow \mathcal{U} \setminus M$
for $iteration \leftarrow 1$ to $setting$ do
    $\Theta \leftarrow$ fine-tuned with $\mathcal{L}$
    $score\_basic \leftarrow$ computed with $\Theta$, $\mathcal{L}$, $\mathcal{U}$ by Eq. 8
    select label-pairs $s$ by $score\_basic$, $S \leftarrow S \cup s$
    while $|S| < setting$ do
        $\overline{\mathbf{X}} \leftarrow$ compute center feature vector of $S$
        $score\_updated \leftarrow$ combine $score\_basic$ & feature distance by Eq. 9
        select label-pairs $s$ by $score\_updated$, $S \leftarrow S \cup s$
    end
    The oracle annotates example-label pairs in $S$
    Update data pool, $\mathcal{L} \leftarrow \mathcal{L} \cup S$, $\mathcal{U} \leftarrow \mathcal{U} \setminus S$, $S \leftarrow \emptyset$
end
```

4 Experiment

4.1 Datasets

Toxic Comment.[3] A multi-label text dataset created by Jigsaw and Google. This dataset is used to evaluate how well machine learning algorithms can detect and filter harmful comments. The comments in the dataset come from Wikipedia discussion pages, and each comment is labeled as zero or more of 6 categories, including toxic, severe_toxic, obscene, threat, insult, and identity_hate.

EurLex.[4] A subset of the EurLex57k dataset[5] which contains 57k legislative documents in English from EUR-Lex with an average length of 727 words. We utilize the reduced version organized in [17]. This version excluded classes with less than 50 positive samples and removed a few texts with no labels.

Go_emotions.[6] A emotional dataset curated from Reddit comments and each text is labeled for 28 emotion categories. To address the imbalance between positive and negative samples, we followed the organization in [17] and retrained the top 10 categories with the largest number of positive samples.

Table 2 provides a comprehensive overview of various statistical features. Based on the label cardinality value, which is below 1, a considerable amount of text in the Toxic Comment and Go_emotions datasets is not assigned to any of the given labels. Besides, these datasets exhibit a significant class imbalance, with the highest ratio of positive to negative samples in Toxic Comment reaching 1:307.

Table 2. Statistic features of datasets

Dataset		Size		Label	Label	Label	Max unba-
	train	test	dev	num	cardinality	density	lance ratio
Toxic Comment	89359	38297	31915	6	0.2178	0.0363	1 : 307
EurLex	10293	1904	1900	10	1.4850	0.1485	1 : 15
Go_emotions	20000	5000	5000	10	0.7616	0.762	1 : 27

4.2 Experimental Details

Compared Methods. We evaluate the performance of our proposed MCVIE through comparing three related and representative MLALs and random sampling. To be more specific, Entropy [17] selects example-label pairs with the

[3] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data.

[4] https://github.com/strwberry-smggls/ActiveLearningTextClassification/tree/main/AL/datasets.

[5] https://github.com/iliaschalkidis/lmtc-eurlex57k/tree/master/data/datasets.

[6] https://huggingface.co/datasets/go_emotions.

highest uncertainty. CVIRS [10] queries the most informative examples by incorporating two uncertainty measures. And CBMAL [20] selects example-label pairs with high informativeness and low redundancy by adopting a two-phase selection. Additionally, we use the full training data to obtain the full performance as a observation.

Implementation Details. To acquire more accurate evaluation and better restore the application scenario of multi-label texts, we employ a pre-trained distilBERT as the base classifier, which consists of a simple linear, ReLu and 0.1 dropout layer. The max length of input texts is 512 and the batch size of training is 16. Besides, we train the distilBERT from scratch on a NVIDIA V100 using Adam optimizer with learning rate of 1×10^{-5}. The max epochs is set to 10 for early stopping and we evaluate the Macro F1 Score of development data to find the best save point of the model and acquire the variable threshold τ that will be mentioned in Subsect. 4.3. Besides, we set 20 iterations including initialization and annotate 50 example-label pairs for every class in each iteration.

4.3 Evaluation Metrics

Unlike multi-class classification where a text is assigned to the class with the highest predicted probability, it is not immediately clear which probabilities indicate a positive prediction in multi-label scenario. To this point, we define a variable threshold τ using Eq. 2 to obtain the observations from the classifier. τ is not fixed as a hyperparameter but can be learned from the model update when using dev set multiple times to calculate the generalization ability of the model. In this process, we set τ to be variable within a certain range, with a step size of t. Every τ in the range is used as a probability boundary, and the corresponding Macro F1 score is calculated. We record the τ value that yields the best Macro F1 score and pass it the test process of the model. The τ is learned as:

$$\tau = argmax\{\text{Macro F1}_1, \text{Macro F1}_2, ...\},\ \tau \in [\min, \max],\ step = t \tag{10}$$

Moreover, to ensure a more accurate measurement for imbalanced fact in multi-label text classification and prevent the model from being overly biased towards negative samples, we utilize the Macro F1 and Micro F1 Score as the evaluation metrics, which are computed as:

$$\text{Macro F1}\,(y, y^*) = \frac{1}{|C|}\sum_{c \epsilon C} \text{F1}_c,\ \text{F1}\,(y, y^*) = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{11}$$

$$\text{Micro F1}\,(y, y^*) = \frac{\sum_{c \epsilon C} TP_c}{\sum_{c \epsilon C} TP_c + \frac{1}{2}(\sum_{c \epsilon C} FP_c + \sum_{c \epsilon C} FN_c)} \tag{12}$$

where y^* are the observations from the classifier and y are the ground truth. TP, FP, FN separately denote true positive, false positive, false negative from the confusion matrix computed by y^* and y.

4.4 Result and Analysis

As shown in Fig. 2, our proposed MCVIE outperforms the compared methods on both the Toxic and Eurlex datasets when the same number of example-label pairs are selected and annotated. Specifically, on the Toxic Comment dataset, our method shows superiority in overall performance as it outperforms other methods on the toxic dataset in terms of Macro F1 Score. Although the Micro F1 Score is closed to that of other methods, our approach still possesses slight advantages. As for the Eurlex dataset, our MCVIE still outperforms others, but the CBMAL is highly competitive, with its curves closely approaching ours. Regarding the Go_emotioms dataset, all methods show a fluctuating score trend, especially for Micro F1 Score. However, the curve of our approach is relatively better and more stable.

In the early stages of MLAL, the random selection can often be highly effective. This is probably because, with limited training data available, MLAL can experience a "cold-start" problem, where the algorithm lacks sufficient information to make informed selections. In such cases, the random selection can be helpful since it aligns with the underlying distribution of the dataset. However, as the number of iterations increases, MLAL strategies tend to outperform the random selection by a significant margin.

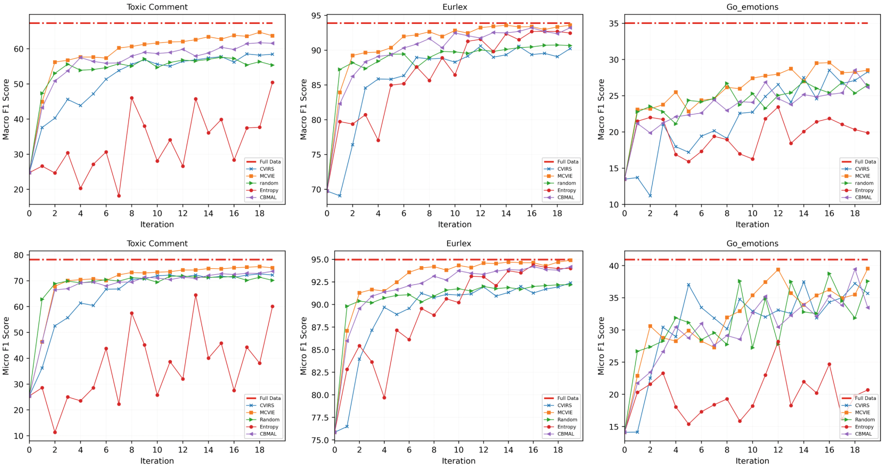


Fig. 2. Macro F1 and Micro F1 scores for all datasets. The x-axis represents the number of iterations in the MLAL process, while the y-axis shows the corresponding score.

Additionally, Table 3 shows the maximum Macro and Micro F1 Score during the entire MLAL process and the ratio of used data to full data is attached to the side. On all datasets, MCVIE reaches better performance compared to other baselines. Especially on the Toxic Comment dataset, the Macro and Micro F1 Score of our MCVIE reaches 64.70 and 75.46 with 1.15% of full data while the full performance is 67.31 and 78.21, respectively. MCVIE achieves a Macro F1 score nearly 2.5% higher than that of CBMAL, which secured the second place.

Table 3. Max Macro F1 Score and Micro F1 Score during the entire AL process and the corresponding ratio of used data to full data.

Dataset	Method	Max Macro F1 & used data	Max Micro F1 & used data	Full data performance
Toxic Comment	Random	57.59, 9.81‰	71.86, 8.14‰	Ma F1: 67.31 Mi F1: 78.21
	Entropy	50.42, 1.21%	64.47, 8.69‰	
	CVIRS	58.52, 1.09%	72.66, 1.15%	
	CBMAL	62.26, 1.06%	73.66, 1.06%	
	MCVIE	**64.70**, 1.15%	**75.46**, 1.15%	
Eurlex	Random	90.75, 11.00%	92.19, 11.48%	Ma F1: 93.90 Mi F1: 94.97
	Entropy	92.71, 10.51%	94.37, 10.03%	
	CVIRS	90.60, 8.08%	92.40, 11.48%	
	CBMAL	93.22, 11.48%	94.21, 10.03%	
	MCVIE	**93.59**, 11.48%	**94.91**, 11.48%	
Go_emotions	Random	26.98, 4.87%	38.75, 5.37%	Ma F1: 35.03 Mi F1: 40.94
	Entropy	23.45, 4.37%	28.19, 4.37%	
	CVIRS	28.52, 5.37%	37.47, 4.87%	
	CBMAL	28.53, 5.87%	39.44, 5.87%	
	MCVIE	**29.57**, 5.37%	**39.53**, 6.12%	

5 Conclusion

We propose an efficient active learning method called MCVIE to reduce the amount of annotation effort required for multi-label text data.[7] Experiments conducted on three public multi-label text datasets demonstrate the effectiveness of the query strategy designed in MCVIE. For future work, we aim to investigate whether our method is equally effective when applied to multi-label image data. Besides, the computation of selecting samples will be further reduced.

References

1. Cherman, E.A., Papanikolaou, Y., Tsoumakas, G., Monard, M.C.: Multi-label active learning: key issues and a novel query strategy. Evol. Syst. **10**, 63–78 (2019)
2. Culotta, A., McCallum, A.: Reducing labeling effort for structured prediction tasks. In: AAAI, vol. 5, pp. 746–751 (2005)
3. Gonçalves, T., Quaresma, P.: A preliminary approach to the multilabel classification problem of Portuguese juridical documents. In: Pires, F.M., Abreu, S. (eds.) EPIA 2003. LNCS (LNAI), vol. 2902, pp. 435–444. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24580-3_50
4. Gui, X., Lu, X., Yu, G.: Cost-effective batch-mode multi-label active learning. Neurocomputing **463**, 355–367 (2021)

[7] This work was completed during the internship at China Mobile.

5. Li, X., Wang, L., Sung, E.: Multilabel SVM active learning for image classification. In: 2004 International Conference on Image Processing, ICIP 2004, vol. 4, pp. 2207–2210. IEEE (2004)
6. Mujawar, S.S., Bhaladhare, P.R.: An aspect based multi-label sentiment analysis using improved BERT system. Int. J. Intell. Syst. Appl. Eng. **11**(1s), 228–235 (2023)
7. Nadeem, M.I., et al.: SHO-CNN: a metaheuristic optimization of a convolutional neural network for multi-label news classification. Electronics **12**(1), 113 (2022)
8. Parvaneh, A., Abbasnejad, E., Teney, D., Haffari, G.R., Van Den Hengel, A., Shi, J.Q.: Active learning by feature mixing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12237–12246 (2022)
9. Rafi, M., Abid, F.: Learning local and global features for optimized multi-label text classification. In: 2022 International Arab Conference on Information Technology (ACIT), pp. 1–9. IEEE (2022)
10. Reyes, O., Morell, C., Ventura, S.: Effective active learning strategy for multi-label learning. Neurocomputing **273**, 494–508 (2018)
11. Reyes, O., Ventura, S.: Evolutionary strategy to perform batch-mode active learning on multi-label data. ACM Trans. Intell. Syst. Technol. (TIST) **9**(4), 1–26 (2018)
12. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
13. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a Core-Set approach. arXiv preprint arXiv:1708.00489 (2017)
14. Shui, C., Zhou, F., Gagné, C., Wang, B.: Deep active learning: unified and principled method for query and training. In: International Conference on Artificial Intelligence and Statistics, pp. 1308–1318. PMLR (2020)
15. Song, R., et al.: Label prompt for multi-label text classification. In: Applied Intelligence, pp. 1–15 (2022)
16. Wang, M., Feng, T., Shan, Z., Min, F.: Attribute and label distribution driven multi-label active learning. Appl. Intell. **52**(10), 11131–11146 (2022)
17. Wertz, L., Mirylenka, K., Kuhn, J., Bogojeska, J.: Investigating active learning sampling strategies for extreme multi label text classification. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 4597–4605 (2022)
18. Wu, J., et al.: Multi-label active learning algorithms for image classification: overview and future promise. ACM Comput. Surv. (CSUR) **53**(2), 1–35 (2020)
19. Wu, K., Cai, D., He, X.: Multi-label active learning based on submodular functions. Neurocomputing **313**, 436–442 (2018)
20. Yang, B., Sun, J.T., Wang, T., Chen, Z.: Effective multi-label active learning for text classification. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 917–926 (2009)
21. Zhang, M., Plank, B.: Cartography active learning. arXiv preprint arXiv:2109.04282 (2021)
22. Zhang, X., Xu, J., Soh, C., Chen, L.: LA-HCN: label-based attention for hierarchical multi-label text classification neural network. Expert Syst. Appl. **187**, 115922 (2022)

Task-Consistent Meta Learning for Low-Resource Speech Recognition

Yaqi Chen, Hao Zhang, Xukui Yang, Wenlin Zhang, and Dan Qu(✉)

School of Information Systems Engineering, Information Engineering University,
Zhengzhou, China
qudanqudan@sina.com

Abstract. We propose a new meta learning based framework that enhances previous approaches for low-resource speech recognition. Meta-learning has proven to be a powerful paradigm for transferring knowledge from prior tasks to facilitate the learning of a novel task. However, when faced with complex task environments and diverse task learning directions, averaging all task gradients is ineffective at capturing meta-knowledge. To address this challenge, we propose a task-consistent multilingual meta-learning (TCMML) method that adopts the gradient agreement algorithm to direct the model parameters in a direction where tasks have more consistency. If a task's gradient matches the average gradient, its weight in meta-optimization is increased, and vice versa. Experiments on two datasets demonstrate that our proposed system can achieve comparable or even superior performance to state-of-the-art baselines on low-resource languages, and can easily combine with various meta learning methods.

Keywords: meta learning · low-resource · speech recognition · task consistent

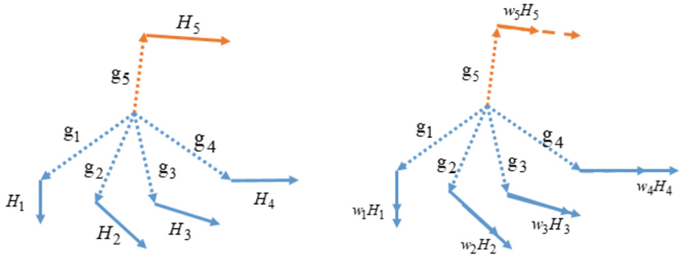
1 Introduction

"Hi, Siri." has featured an enormous boom of smart speakers in recent years, unveiling the success of automatic speech recognition (ASR). As a critical communication bridge between humans and machines, the performance of ASR depends heavily on massively transcribed data, which is labor-intensive and time-consuming to collect. Interestingly, despite the diverse languages spoken worldwide, people often express similar meanings, independent of their symbol systems. Multilingual transfer learning (MTL-ASR) [1] leverages this fact to enhance the performance of low-resource languages by transferring knowledge from rich-resource languages. MTL-ASR constructs a model with shared hidden layers across different languages, which benefits downstream tasks. However, it may be less effective under very-low-resource conditions due to the overfitting problem.

Learning quickly is a hallmark of human intelligence, as evidenced by children's ability to recognize items from only a few examples. Hence, equipping

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 349–360, 2023.
https://doi.org/10.1007/978-3-031-44693-1_28

a deep model with the ability to learn new concepts from a few labeled data is significant for practical use. Recently, meta-learning [2] aims to develop efficient algorithms (e.g., needing a few or even no task-specific fine-tuning) that can learn the new task quickly, providing a promising paradigm to tackle the low-resource problem. Inspired by this, some research has proposed multilingual meta learning (MML-ASR) [3]. MML-ASR trains a well-generalized model initialization that adapts to new task quickly. The key lies in optimizing the initialization's ability to generalize, which is assessed by how well the adapted model performs on the query set for each task. One of the most representative algorithms in this area is model-agnostic meta-learning (MAML) [4], which has shown superior performance in several low-resource fields, including speaker adaptation [5], accent adaptation [6], speech recognition [7], and emotion recognition [8]. In multilingual meta learning, various approaches have been proposed to improve model generalization. For instance, adversarial meta sampling [9] trained an extra sampled network to solve the task imbalance problem. Meta-adaptor [10] was proposed to achieve more efficient fine-tuning, while multi-step weighted loss [11] was adopted to improve the stability of meta learning. As an efficient algorithm, MML-ASR is model-agnostic, making it easy to combine with large-scale pretraining models, like wav2vec [12], Hubert [13], WavLM [14], and others.



(a) MML-ASR

(b) TCMML-ASR

Fig. 1. Comparison between multilingual meta learning (MML) and task-consistent multilingual meta learning (TCMML). Where g_i denotes the support gradient, and H denotes the query second-order gradient. Left: MML averages all H in the meta-update stage. Right: TCMML assigns a weight for each H, decreasing the inconsistent task weight, like task 5.

During the meta-update stage, meta-learning averages the gradient of all tasks' query loss. However, they ignore the task-conflict issue for the various languages. As shown in Fig. 1a, suppose we extract a batch of tasks, of which four are gradient updated in one direction and one in the opposite direction in the task-update stage. In this case, different language tasks will compete for a model's limited resources during the meta-update stage. Inconsistencies between tasks have a negative impact on multilingual meta learning, leading to inefficient learning.

To solve the task-conflict issue, we propose task-consistent multilingual meta learning (TCMML), which employs the gradient agreement algorithm [15] for multilingual meta learning. It assigns a weight for each task during the meta-update stage, which is determined by the inner product of the task gradient and the average gradient across a batch of tasks. As shown in Fig. 1b, if a task's gradient aligns with the average gradient, the weight for that task increases, like task1-4. Conversely, if the task gradient conflicts with the average gradient, the weight decreases, like task 5. In this way, TCMML directs the model parameters in a direction where tasks have more consistency during meta-update stage.

There also exist some works that deal with the task-conflict problem. Yunshu et al. [16] proposed discarding auxiliary tasks when their gradient conflicts with the average task gradient. But we keep these conflict task gradients to extract more language knowledge. PCGrad [17] addresses the task-conflict problem through projection. However, it needs to calculate the cosine similarity between each pair of tasks for projection, which requires a large amount of computation, and changing the inner gradient may cause the mismatch between task-learn and meta-learn stage. Moreover, Guiroy et al. [18] proposed a regularized MAML with an angle penalty between inner loop updates, which is more expensive as it requires repeating the inner loop for all languages. But our TCMML only need to assign a weight to different tasks in meta-update stage, which is more simpler.

We conducted experiments on four low-resource languages from IARPA BABEL and seven very-low-resource languages from OpenSLR to evaluate the effectiveness of TCMML. The results show that TCMML can effectively improve the performance of all low-resource languages. Moreover, TCMML can enhance model performance with various meta learning methods.

2 Preliminaries

2.1 The ASR Model

Our multilingual speech recognition model utilizes the joint CTC-attention [19] architecture, which has also been used in previous studies [9,20]. As illustrated in Fig. 2, the model consists of three parts: the encoder and the decoder in the Transformer, as well as a connectionist temporal classification (CTC) module, which can guide the model to learn good alignments and speed up convergence. The loss function can be expressed as a weighted sum of the decoding loss L_{att} and the CTC loss L_{ctc}:

$$L = \lambda L_{ctc} + (1 - \lambda) L_{att}, \tag{1}$$

where the hyper-parameter λ denotes the weight of CTC loss. To overcome the challenge posed by different language symbols, BPE [21] is employed to generate sub-words as multilingual modeling units. Training transcripts from all languages

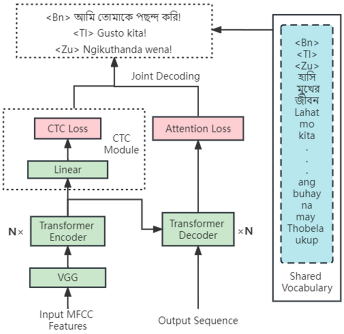


Fig. 2. Overview of the Transformer ASR model based on hybrid CTC-attention architecture.

are combined together to generate the multilingual symbol vocabulary. As a result, similar sub-words are shared among different languages, which is very beneficial for similar languages to learn common information. Similar to prior work [22], we substitute the language symbol $<S\_LANG>$ for the start token $<S>$ at the beginning of the original sub-word sequence, which can alleviate the language confusion.

2.2 Multilingual Meta Learning

In contract to traditional machine learning, meta learning uses tasks (or episodes) as its training sample. In order to learn target languages, multilingual meta learning has to acquire generic meta-knowledge over a number of training episodes. Suppose the dataset is a set of N languages $D_s = D_s^i(i = 1, ...N)$, each language D_s^i is composed of the speech-text pairs. For i-th language, we sample tasks T_i from the $D_s^i(i = 1, 2, ..., N)$, and divide T_i into two subsets, the support set T_{sup}^i and the query set T_{query}^i. The ASR model f_θ is parameterized by θ. After multilingual pretraining, the model aims to adapt to the low-resource target languages D_t. Learning valuable meta-knowledge from D_s is crucial for meta-learning models. Present gradient-based meta-learning techniques can be described as bilevel optimization problems, with the episodic training paradigm applied to train the entire model. The two-level meta-learning framework can be characterized as:

$$\min_{\theta} \sum_{i=1}^{N} L^{meta}(\theta, \omega^{*(i)}(\theta); T_{query}^i) \tag{2}$$

$$s.t. \;\; \omega^{*(i)}(\theta) = \arg\min l(\omega; \theta, T_{\text{sup}}^i) \tag{3}$$

where L^{meta} and l refer to the meta loss (in the outer loop) and the task loss (in the inner loop), respectively.

In particular, the inner loop (Eq. (2)) is designed to learn a language-specific base learner for each individual task using the support set T_{sup}^i, whereas the outer loop (Eq. (3)) learns meta-knowledge from these base learners with the query set T^i_{query}, which can be utilized to adapt to new language quickly. As benchmarks for our research, we then describe three common two-level gradient-based meta learning techniques.

MAML: MAML [4] is one of the most representative algorithms in this area, utilized across various low-resource fields. MAML aims to generalize across the task distribution by learning a better initialization θ. We can formulate the meta-learning process as:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L^{meta}(\omega^{(i)}(\theta); T^i_{query}) \tag{4}$$

$$s.t.\ \ \omega^{(i)}(\theta) = \theta - \alpha \nabla_{\theta} l(\theta; T^i_{\text{sup}}) \tag{5}$$

where α is the learning rate of the base learner and l is the task loss function. Equation (5) means one step of the inner loop to learn task T_i.

FOMAML: FOMAML is often used as an alternative to MAML due to the challenge of computing the second-order derivative and storing the Hessian matrix. For instance, [9] and [20] have utilized FOMAML instead of MAML.

ANIL: Raghu et al. [23] proposed ANIL, a simplification of MAML by leaving only the (task-specific) head of the neural network in the inner loop. This approach has demonstrated comparable performance to MAML.

3 Task-Consistent Multilingual Meta Learning

According to Eq. (2), it can be found that meta-learning averages all tasks' query gradient to update model parameters, indicating that all tasks contribute equally to updating model parameters. However, the difference of various languages will lead the task-conflict problem, causing inefficient learning as shown in Fig. 1. We adopt the gradient agreement algorithm to adjust each task's contribution to the overall gradient updating direction. Specifically, the formula for parameter updating during the outer loop is:

$$\theta \leftarrow \theta - \beta \sum_{i} \omega_i \nabla L^{meta}(\omega^{(i)}(\theta); T^i_{query}) \tag{6}$$

where w_i denotes the weight assigned to the i-th task during meta-update stage. Specifically, we decrease the weight of tasks whose updating direction deviates from that of the majority of tasks, and increase the weight of tasks whose updating direction aligns with that of most tasks. Let the full-model parameter gradient of each task be represented by g_i, and $g_i = \theta_i - \theta$. We introduce the average gradient across all tasks g_v, which can be defined as:

$$g_v = \frac{1}{N} \sum_{i}^{N} g_i \tag{7}$$

Algorithm 1. Task-Consistent Multilingual Meta Learning

```
Require: ASR model $f$ parameterized by $\theta$; Learning rates of $\alpha$ and $\beta$;
Initialize ASR model $f_\theta$
while not done do
    Sample batch of tasks $T_i \sim p(T)$
    for all $T_i$
    Generate support set $T^i_{sup}$ and query set $T^i_{query}$ from $T_i$
    Construct ASR support loss $L_{T^i_{\mathrm{sup}}}$ on $T^i_{sup}$ using Eq. (1)
    Compute adapted parameters $\theta_i$ using Eq. (5)
    Compute $g_i$ using $g_i = \theta_i - \theta$
    Construct ASR query loss $L_{T^i_{query}}$ on $T^i_{query}$ usimg Eq. (1)
    end for
    Compute $g_v$ using Eq. (7)
    Compute $w_i$ using Eq. (8)
    Update model parameters using Eq. (6)
end while
```

If the i-th task gradient g_i in conflict with the average gradient of all tasks g_v, the i-th task should contribute less than other tasks in the meta-update stage. The w_i needs be proportional to the inner product of the task gradient and the average gradient of all tasks in a batch $g_i^T g_v$. Moreover, the weight needs to satisfy $\sum\limits_i \omega_i = 1$. So the weight w_i can be defined as:

$$\omega_i = \frac{g_i^T g_v}{\sum\limits_{k \in N} \left| g_k^T g_v \right|} = \frac{\sum\limits_{j \in N} g_i^T g_j}{\sum\limits_{k \in N} \left| \sum\limits_{j \in N} g_k^T g_j \right|} \tag{8}$$

In this way, if the task gradient aligns with the average gradient of all tasks, its weight w_i increases. If not, its weight decreases. With this insight, the proposed optimization method, pushes the model parameters in a direction where tasks have more consistency during meta-update stage. We summarize our whole algorithm in Algorithm 1.

4 Experiment

4.1 Dataset

Our experiments are based on the IARPA BABEL [24] and OpenSLR[1]. Specifically, IARPA BABEL consists of conversational telephone speech in 25 languages, collected in various environments. Each language contains a Full Language Pack (FLP) and a Limited Language Pack (LLP). We constructed two datasets: Babel3 and Babel6. To construct Babel3, we selected three languages:

[1] https://openslr.org/resources.php.

Bengali (Bn), Tagalog (Tl), and Zulu (Zu). To construct Babel6, we selected six languages: Bengali (Bn), Tagalog (Tl), Zulu (Zu), Turkish (Tr), Lithuanian (Lt), and Guarani (Gn). The pre-trained model is trained on the FLP of the language. We also selected four languages as target languages: Vietnamese (Vi), Swahili (Sw), Tamil (Ta), and Kurmanji (Ku). We fine-tuned the model by using its FLP, and tested it using LLP (about 10 h). The OpenSLR dataset is an open-source collection of speech and language resources and contains different languages and accents, including low-resource languages. For OpenSLR dataset, we selected nine languages as source languages for pre-training (OpenSLR-9), and fine-tuned the model on seven very-low-resource target languages: Argentinian Spanish (SLR-61), Malayalam (SLR-63), Marathi (SLR-66), Nigerian English (SLR-70), Venezuelan Spanish (SLR-75), Burmese (SLR-80), and Yoruba (SLR-86). For the pretraining in OpenSLR, we used 80% of its data to train and 20% to validate. For each target language, we used 60% of its data to train, 10% to validate, and 30% to test. Table 1 describes the dataset statistics for the experimental data.

Table 1. Multilingual dataset statistics in terms of hours (h) languages.

BABEL	Source	Bengali	61.76	Tagalog	84.56	Zulu	62.13
		Turkish	77.18	Lithuanian	42.52	Guarani	43.03
	Target	Vietnamese	87.72	Swahili	44.39		
		Kurmanji	42.08	Tamil	68.36		
OpenSLR	Source	Gujarati	7.89	Colombian Spanish	7.58	English	5.00
		Tamil	7.08	Peruvian Spanish	9.22	Galician	10.31
		Kannada	8.68	Chilean Spanish	7.15	Basque	13.86
	Target	Burmese	4.11	Argentinian Spanish	8.08	Yoruba	4.02
		Marathi	2.98	Nigerian English	2.32		
		Malayalam	2.49	Venezuelan Spanish	2.40		

4.2 Network Configurations

For IARPA BABEL, the model uses a 6-layer VGG convolutional network, and the Transformer consists of 4 encoder blocks and 2 decoder blocks. Each block comprises four attention heads, 512 hidden units, and 2048 feed-forward hidden units. The model is trained using a batch size of 128, with 64 examples allocated to the support set and 64 examples to the query set. The experiments are conducted using two Tesla V100 16 GB GPUs. The SGD algorithm is used for the task-update stage in meta-learning, while Adam optimizer is used for the rest of the optimization. We set warmup steps to 12000 and k to 0.5. For OpenSLR-9, the model uses one encoder block and one decoder block of the Transformer architecture. The model is trained using a batch size of 96, with 48 examples allocated to the support set and 48 examples to the query set. We set warmup steps to 1000 and k to 0.5. The other settings are the same as IARPA BABEL. During fine-tuning target languages, we substitute the output layer and vocabulary. The batch size is set to 128, and other settings are the same as pre-training.

4.3 Training and Evaluation

We used the Kaldi [25] toolkit for feature extraction to obtain 40-dimensional Mel-frequency Cepstral Coefficients (MFCC) features and 3-dimensional pitch features computed every 10 ms over a 25 ms window for both IARPA BABEL and OpenSLR. And we adopted SpecAugment [26] for data augmentation. We evaluated our model using beam-search with a beamwidth of 6 and the weight for CTC loss $\lambda = 0.3$. In the experiments, character error rate (CER) is employed as the criterion, and the five best models are saved during training. We trained 200 epochs and set the early stop strategy until the criterion remains unchanged for three times. Finally, we averaged the five best model parameters as the final model.

5 Experiment Results

5.1 Results on Low-Resource ASR

Results on OpenSLR. In this study, we analyzed the impact of meta-learning on target languages under very-low-resource conditions. Table 2 reports the results on OpenSLR in terms of character error rate (CER). Firstly, Monolingual is very poor due to a lack of knowledge of other languages. Secondly, both MTL-ASR and MML-ASR outperform Monolingual on all target languages, which can effectively improve the performance of low resource languages. And MML-ASR decreased CER over MTL-ASR by 5% due to its fast learning ability by learning a better initialization. Finally, our TCMML further improves MML-ASR by over 2% for all target languages.

Table 2. Results of low resource ASR on OpenSLR-9 in terms of CER (%)

Methods	SLR-61	SLR-63	SLR-66	SLR-70	SLR-80	SLR-75	SLR-86	Avg.
Monolingual	50.24	80.95	70.39	95.07	74.95	68.49	76.82	73.84
MTL-ASR	42.33	53.29	47.33	65.92	58.25	48.01	61.42	53.79
MML-ASR	38.62	44.93	39.59	62.73	49.16	36.28	57.78	47.01
TCMML	**34.91**	**42.05**	**37.31**	**60.64**	**47.87**	**35.05**	**52.91**	**44.39**

Results on IARPA BABEL. To further verify the effectiveness of our proposed TCMML, we conducted experiments to compare with earlier works [3,9] on the IARPA BABEL dataset. Table 3 reports the CER results on IARPA BABEL for four target languages. Compared to work in [3], our results perform better because the joint CTC-attention structure is better than a CTC model. Moreover, our baseline performance is similar to the joint CTC-attention structure used in [9], although there are differences in the encoder and decoder structures. But our model size is also similar to that of [9], which has approximately 29M and 26M parameters, respectively. Finally, by pushing the model parameters in a direction of more consistent tasks, our TCMML-ASR outperforms MML-ASR for all languages. Moreover, in general, our method achieves better performance than meta adversarial sampling(AMS) in [9] and our TCMML-ASR is simpler for it doesn't require any extra model components.

Table 3. Results of low resource ASR on IARPA BABEL in terms of CER (%)

Target	Vietnamese		Swahili		Tamil		Kurmanji	
Source	Babel3	Babel6	Babel3	Babel6	Babel3	Babel6	Babel3	Babel6
MTL-ASR [3]	57.4	59.7	48.1	48.8	65.6	65.6	67.6	66.7
MML-ASR [3]	49.9	50.1	41.4	42.9	57.5	58.9	64.6	64.1
MML-ASR [9]	/	45.1	/	36.14	/	50.16	/	/
AMS-ASR [9]	/	43.35	/	32.19	/	48.56	/	/
Monolingual	69.06		57.31		59.08		73.76	
MTL-ASR	43.84	42.03	37.63	35.47	51.54	50.84	60.90	57.00
MML-ASR	43.99	42.07	33.57	32.38	51.58	48.47	59.70	58.62
TCMML	**42.96**	**41.65**	**32.51**	**30.84**	**48.87**	**47.76**	**58.06**	**57.44**

In summary, we evaluated our model on different quality data, such as OpenSLR which has high-quality data and IARPA BABEL which has more noise. We also varied the number of languages and amount of data, ranging from Babel3, Babel6 to OpenSLR-9, with target language data ranging from about 2 h to 60 h. In these different experimental settings, TCMML achieved significant improvements in the performance of all target languages, demonstrating better generalization and robustness.

Visualization of Model Learning Dynamic. The dynamics of the model training loss are depicted in Fig. 3. The blue and yellow curves represent the accuracy of MML-ASR and TCMML-ASR on the validation set, respectively. It can be observed that our convergence epoch can be improved by over 30% from about 100 epochs to 70 epochs. Moreover, we evaluated the computational cost incurred by TCMML. We found that without using TCMML, the average time required per iteration was 4.57 s, while with TCMML it took 5.08 s. So introducing TCMML leads to approximately a 10% increase in average time required per epoch for computational cost. Therefore, TCMML demonstrates faster learning speed by over 20 % overall compared to MML.

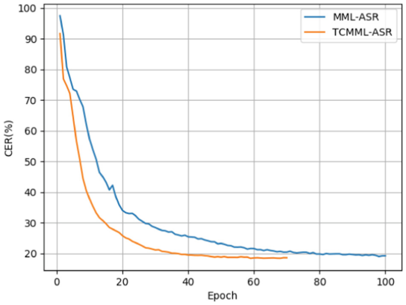


Fig. 3. Visualization of training process.

5.2 Ablation Studies

Different Meta Learning Methods. We also analyzed the performance of different gradient-based meta-learning methods, such as MAML, FOMAML, and ANIL, on three low-resource languages in OpenSLR: SLR-61, SLR-80 and SLR-86. Table 4 indicates that TCMML can improve all of the meta learning methods, especially in FOMAML. Among them, TCMML-ANIL performs worse than other methods. This can be attributed to the fact that its weights are determined by the inner gradient of model parameters, while ANIL only modifies the output layer in the inner loop. As a result, the weight calculated in the outer loop is not correct, leading to suboptimal performance.

Table 4. Ablation study results on OpenSLR in terms of CER (%)

Methods	SLR-61	SLR-80	SLR-86	Avg.
Monolingual	50.24	74.95	76.89	67.36
MTL-ASR	42.33	58.25	56.52	52.37
MML-ASR (FOMAML)	38.62	49.16	53.24	47.00
MML-ASR (MAML)	36.97	**47.63**	**52.42**	**45.67**
MML-ASR (ANIL)	**36.38**	49.36	54.04	46.59
TCMML-FOMAML	**34.91**	**47.87**	50.14	44.30
TCMML-MAML	35.05	48.04	**48.25**	**43.78**
TCMML-ANIL	36.26	48.80	50.52	45.19
TCMML-FOMAML (80%-target)	35.71	49.38	55.54	**46.87**
TCMML-FOMAML (50%-target)	38.26	54.35	61.17	51.26
TCMML-FOMAML (20%-target)	43.18	66.66	71.98	60.60

Different Scales of Training Data. To verify the effectiveness of our proposed TCMML when data is limited, we evaluated the performance of TCMML using only 80%, 50%, and 20% of the training data. As shown in Table 4, even with 80% of the training data, TCMML still outperforms most baseline systems, demonstrating that TCMML can effectively reduce the dependency on heavily annotated data.

6 Conclusion

In this work, to tackle the task-conflict problem, we propose a task-consistent multilingual meta learning (TCMML) algorithm for low-resource speech recognition by directing model parameters in a consistent direction. Extensive experimental results demonstrate that our method effectively enhances the few-shot learning ability of meta-learning. In the future, we plan to exploit the task gradient adjusting strategy adaptively for acquiring more knowledge in meta-learning.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 62171470),Henan Zhongyuan Science and Technology Innovation Leading Talent Project (No. 234200510019), and Natural Science Foundation of Henan Province of China (No. 232300421240).

References

1. Luo, J., Wang, J., Cheng, N., Zheng, Z., Xiao, J.: Adaptive activation network for low resource multilingual speech recognition. In: IJCNN, pp. 1–7 (2022)
2. Hospedales, T.M., Antoniou, A., Micaelli, P., Storkey, A.J.: Meta-learning in neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 5149–5169 (2020)
3. Hsu, J.-Y., Chen, Y.-J., Lee, H.-Y.: Meta learning for end-to-end low-resource speech recognition. In: ICASSP 2020, pp. 7844–7848. IEEE (2020)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Precup, D., Teh, Y.W. (eds.) ICML, vol. 70, pp. 1126–1135 (2017)
5. Klejch, O., Fainberg, J., Bell, P., Renals, S.: Speaker adaptive training using model agnostic meta-learning. In: ASRU, pp. 881–888 (2019)
6. Winata, G.I., Cahyawijaya, S., Liu, Z., Lin, Z., Madotto, A., Xu, P., Fung, P.: Learning fast adaptation on cross-accented speech recognition. In: Meng, H., Xu, B., Zheng, T.F. (eds.) Interspeech, pp. 1276–1280 (2020)
7. Naman, A., Deepshikha, K.: Indic languages automatic speech recognition using meta-learning approach. In: InterSpeech (2021)
8. Chopra, S., Mathur, P., Sawhney, R., Shah, R.R.: Meta-learning for low-resource speech emotion recognition. In: ICASSP, pp. 6259–6263 (2021)
9. Xiao, Y., Gong, K., Zhou, P., Zheng, G., Liang, X., Lin, L.: Adversarial meta sampling for multilingual low-resource speech recognition. In: AAAI (2020)
10. Hou, W., Wang, Y., Gao, S., Shinozaki, T.: Meta-adapter: efficient cross-lingual adaptation with meta-learning. In: ICASSP, pp. 7028–7032 (2021)
11. Singh, S., Wang, R., Hou, F.: Improved meta learning for low resource speech recognition. In: ICASSP, pp. 4798–4802 (2022)
12. Baevski, A., Zhou, H., Mohamed, A.R., Auli, M.: Wav2vec 2.0: a framework for self-supervised learning of speech representations. ArXiv, abs/2006.11477 (2020)

13. Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.R.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
14. Chen, S., et al.: WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Top. Signal Process. **16**, 1505–1518 (2021)
15. Eshratifar, A.E., Eigen, D., Pedram, M.: Gradient agreement as an optimization objective for meta-learning. CoRR, abs/1810.08178 (2018)
16. Du, Y., Czarnecki, W.M., Jayakumar, S.M., Pascanu, R., Lakshminarayanan, B.: Adapting auxiliary losses using gradient similarity. ArXiv, abs/1812.02224 (2018)
17. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: Larochelle, H., Ranzato, M.A., Hadsell, R., Balcan, M.-F., Lin, H.-T. (eds.) NeurIPS 2020 (2020)
18. Guiroy, S., Verma, V., Pal, C.: Towards understanding generalization in gradient-based meta-learning. ArXiv, abs/1907.07287 (2019)
19. Kim, S., Hori, T., Watanabe, S.: Joint CTC-attention based end-to-end speech recognition using multi-task learning. In: ICASSP, pp. 4835–4839 (2016)
20. Hou, W., Zhu, H., Wang, Y., Wang, J., Qin, T., Renjun, X., Shinozaki, T.: Exploiting adapters for cross-lingual low-resource speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 317–329 (2021)
21. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL. The Association for Computer Linguistics (2016)
22. Zhou, S., Xu, S., Xu, B.: Multilingual end-to-end speech recognition with a single transformer on low-resource languages. ArXiv, abs/1806.05059 (2018)
23. Raghu, A., Raghu, M., Bengio, S., Vinyals, O.: Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. In: ICLR (2020)
24. Gales, M.J.F., Knill, K.M., Ragni, A., Rath, S.P.: Speech recognition and keyword spotting for low-resource languages: babel project research at CUED. In: SLTU, pp. 16–23. ISCA (2014)
25. Povey, D., et al.: The kaldi speech recognition toolkit (2011)
26. Park, D.S., et al.: SpecAugment: a simple data augmentation method for automatic speech recognition. In: Interspeech (2019)

An Adaptive Learning Method for Solving the Extreme Learning Rate Problem of Transformer

Jianbang Ding[1](✉), Xuancheng Ren[1], and Ruixuan Luo[2]

[1] MOE Key Laboratory of Computational Linguistics,
Peking University, Beijing, China
{jianbangding,renxc,luoruixuan97}@pku.edu.cn

[2] Center for Data Science, Peking University, Beijing, China

Abstract. Transformer, a neural sequence model entirely based on attention, has achieved great success in natural language processing and become the *de facto* default model for multiple NLP tasks. Albeit its prevalence, the attention-based structure poses unmet challenges that the widely-used adaptive optimization methods, e.g., Adam, have serious difficulty in learning and often fail to converge if applied alone. In this work, we illustrate the problem that the adaptive optimization methods produce extremely-large learning rates that break the balance of stability. We further propose AdaMod, which smooths out extremely-large learning rates with adaptive and momental upper bounds on a per-parameter basis, instead of the uniform scaling in the warmup scheme. We empirically demonstrate AdaMod can improve the learning stability and bring significant improvements to the performance of Transformers and CNNs. Moreover, empirical results verify its effectiveness and robustness across different applications.

Keywords: Transformer · Optimization · AdaMod

1 Introduction

Gradient-based optimization forms the core of first-order optimization algorithms to train deep networks. Remarkably, stochastic gradient descent (SGD) [18], one of the most dominant methods, performs well across many applications, despite its simplicity. However, one shortcoming of SGD is that it scales the gradient uniformly in all directions. This strategy requires a subtle tuning of the learning rate and limits the training speed in the early stage. To address this issue, several adaptive methods have been proposed to achieve faster convergence by computing individual learning rates for different parameters. Examples of such methods include AdaGrad [3], Adam [7], RMSProp [20] and AdaDelta [25]. In particular, Adam is regarded as the default algorithm used across many frameworks [23].

Although adaptive methods have gained great popularity, they still stumble on the stability problem of complex models. It has been observed that they may

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-44693-1_29.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 361–372, 2023.
https://doi.org/10.1007/978-3-031-44693-1_29

converge to bad local optima when training Transformers and have to resort to the warmup scheme [16,21], which starts with small learning rates for all the parameters until the system reaches stability and is hence used as a common heuristic in practice [5]. However, different parameters play different roles in the model so scaling all the parameters uniformly in the early training stage is not the best. The warmup scheme also introduces some additional hyperparameters and needs tedious fine-tuning. Recent work has put forward some algorithms such as AMSGrad [17] and RAdam [10] to tackle this issue but they failed to achieve considerable improvement over existing methods. In this paper, we first illustrate that the extremely-large learning rates lead to the non-convergence problem and expect to bound large learning rates on a per-parameter basis throughout the training stage.

Under this premise, we propose a new variant of Adam, that is AdaMod, to restrict the adaptive learning rates with adaptive and momental bounds. This endows learning rates with "long-term-memory" historical statics. With this framework, we can balance the gradients across layers to improve learning stability. Finally, we conduct further experiments on Transformers. Empirical results demonstrate that our method can effectively eliminate unexpected large learning rates and hence fix the non-convergence problem. Moreover, it can bring consistent and significant improvement over the vanilla Adam across different architectures such as ResNet and DenseNet.

The contributions are summarized as follows:

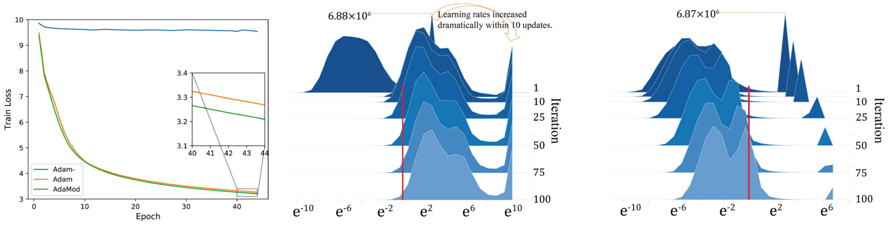
- We propose AdaMod, which eliminates extreme learning rates with adaptive momental bound on a per-parameter basis and brings significant improvements to the performance of Transformers without warmup.
- Experiments demonstrate that AdaMod can be used in various applications such as machine translation, document summarization, and image classification due to its robustness and effectiveness.

2 Background

2.1 A Brief Review of Adam

Algorithm 1. Adam

```
Input: initial parameter $\theta_0$, step sizes $\{\alpha_t\}_{t=1}^T$, first moment decay $\beta_1$, second moment
    decay $\beta_2$, stochastic objective function $f(\theta)$
Initialize $m_0 = 0$, $v_0 = 0$
for $t = 1$ to $T$ do
    $g_t = \nabla f_t(\theta_{t-1})$
    $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ and $V_t = \text{diag}(v_t)$
    $\theta_t = \theta_{t-1} - \alpha_t m_t / \sqrt{V_t}$
end for
```



(a) Training loss (b) Learning rates for Adam w/o warmup (c) Learning rates for Adam with warmup

Fig. 1. Training loss and learning rate distribution of Transformers on the IWSLT'14 De-En dataset. "Adam-" in (a) denotes Adam without warmup. For (b) and (c), X-axis is the original value in the log scale; Y-axis is training iterations and the height stands for frequency. Red line to show 0 centerlines. Adam fails to converge without warmup due to extremely-large learning rates, while AdaMod can fix this issue and perform better. (Color figure online)

Algorithm 1 provides a brief review of Adam for reference, and for brevity, bias-correction operations are not included. The setup is elaborated as follows. We first compute the gradient g_t of the loss function with respect to previous parameters. Second, we update the low-order moments of gradient m_t, v_t by adopting exponential averaging and computing bias-corrected versions for them. Finally, we refresh the parameter to get a new θ_t. This process needs to iterate T steps until we return the learned parameters. Noting that we refer to $\alpha/\sqrt{V}$ as the learning rate in the paper.

2.2 Extremely-Large Learning Rates Leading to Instability Issues

We first illustrate that the extremely-large learning rates will cause the non-convergence problem. For example, in the NMT experiment in Fig. 1a, the training loss converges to around 9.5 without warmup, and it decreases to below 3.5 after using warmup. In addition, the learning rate histogram is shown in Fig. 1b and Fig. 1c, where the X-axis is the original value in the log scale, Y-axis is the iteration steps and the height stands for frequency. We can observe that without warmup, there are lots of learning rates soaring over 10,000 compared to applying it. Such extremely-large learning rates may lead to the oscillation of the sequence and trap the adaptive method in exceptionally bad local optima. Meanwhile, they cannot help the optimizer escape from that, resulting in a series of non-convergence problems. Similar phenomena are observed in other tasks such as Transformer-XL [2] language modeling.

3 Related Work

Exploring how to tackle the non-convergence issue of adaptive methods is an important research interest of current machine learning research. In recent years, many remarkable works have provided us with a better understanding of this problem with the proposal of different variants of Adam. [17] first indicated that Adam may not converge due to the lack of “long-term-memory” of past gradients and provided a theoretical guarantee of convergence. Following this track, most of the previous studies focused on how to modify the re-scaling term v_t. [26] argued that there exists an inappropriate correlation between g_t and v_t, which may result in unbalanced updates of step size. Therefore, the authors proposed decorrelating them by temporal shifting, i.e. replacing g_t with g_{t-n} for some manually chosen n to calculate v_t. In a similar vein, [6] discussed that the past gradients $\{g_1, ..., g_{t-1}\}$ are more reliable than g_t. And the authors proposed to weigh more of all past gradients when designing v_t. However, these methods do not radically avoid the non-convergence problem in practice due to the existence of unexpected large learning rates. To solve this problem, [19] considered dropping momentum and removing the larger-than-desired updates by selecting a threshold d for update clipping. However, as their main goal is to minimize the memory cost of optimization algorithms, this technique remains less explored and has limited improvement in generalization performance. To this end, [10] proposed automatic variance rectification of the adaptive learning rate based on derivations. [12] implemented a gradual transition from Adam to SGD by employing dynamic bounds on learning rates to avoid extremely-larger ones. However, its bound function is manually designed and the performance relies heavily on the selection of the final learning rate α^* of SGD.

By contrast to the original Transformer with post-norm residual units (Post-Norm), some work [1,22,24] use pre-norm residual units (Pre-Norm) which locate the layer normalization inside the residual units to obtain well-behaved gradients. Although the Pre-Norm architecture can alleviate the non-convergence problem by normalizing the large gradients at initialization, the extreme learning rates are still found by the end of training on ResNet with Pre-Norm [12]. Although these learning rates do not cause serious non-convergence as in Transformers, they do harm the generalization performance.

4 Methods

This section describes the AdaMod method as well as its properties. Concisely, AdaMod casts dynamic upper bounds on the adaptive learning rates that prevent the calculated learning rates from escalating too fast and becoming undesirably larger than what the historical statistics suggest. This controls the variance of the adaptive learning rates and smooths out the larger-than-expected fluctuations, hence getting more stable and reliable learning rates. AdaMod names from **Ada**ptive and **Mo**mental Boun**d**. The pseudocode is provided in Algorithm 2.

4.1 Smoothing Adaptive Learning Rates

Based on Adam, which computes adaptive learning rates with estimates of first and second moments (i.e., expectation and uncentered variance) of the gradients,

Algorithm 2. AdaMod

```
Input: initial parameter $\theta_0$, step sizes $\{\alpha_t\}_{t=1}^T$, first moment decay $\beta_1$, second moment decay$\beta_2$, smoothing coefficient $\gamma$, stochastic objective function $f(\theta)$
Initialize $m_0 = 0$, $v_0 = 0$, $\hat{\eta}_0 = 0$
for $t = 1$ to $T$ do
    $g_t = \nabla f_t(\theta_{t-1})$
    $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ and $V_t = \text{diag}(v_t)$
    $\hat{\eta}_t = \gamma \hat{\eta}_{t-1} + (1 - \gamma)\alpha_t / \sqrt{V_t}$
    $\eta_t = \min(\hat{\eta}_t, \alpha_t / \sqrt{V_t})$
    $\theta_t = \theta_{t-1} - \eta_t \odot m_t$
end for
```

our method further estimates the first-order moments of the individual adaptive learning rates $\alpha_t/\sqrt{V_t}$. Inspired by exponential moving average (EMA) which enjoys popularity in estimating the lower-order moments of the gradients. We do average directly on the learning rates $\alpha_t/\sqrt{V_t}$ computed by Adam. Specifically, we apply the following operation:

$$\hat{\eta}_t = \gamma \hat{\eta}_{t-1} + (1 - \gamma)\alpha_t/\sqrt{V_t}, \tag{1}$$

where $\alpha_t/\sqrt{V_t}$ are the learning rates computed by Adam at step t. Thus, the current momental value $\hat{\eta}_t$ is an interpolation between the previous momental value $\hat{\eta}_{t-1}$ and the current learning rates. Since the average range of the data in the exponential moving average is $1/(1-\gamma)$, which can be proven by evaluating an infinite series, the new hyperparameter γ controls the memory of $\hat{\eta}_t$. For example, when $\gamma = 0.9$ the average range is 10 periods; when $\gamma = 0.999$ the average range is 1,000 periods, so on and so forth. It is worth noting that when $\gamma \to 0$, AdaMod degenerates to Adam.

Equation 1 can be expressed in another version, where the current momental value is an exponentially weighted moving average with discount factor γ:

$$\hat{\eta}_t = (1 - \gamma) \sum_{i=1}^{t} \gamma^{t-i} \cdot \alpha_i/\sqrt{V_i}. \tag{2}$$

This endows the current value $\hat{\eta}_t$ with "long-term-memory" of past values $\alpha_i/\sqrt{V_i}$. For general cases, we set $\hat{\eta}_0 = 0$ and $\gamma = 0.9999$.

4.2 Bounding Adaptive Learning Rates

For the current momental value $\hat{\eta}_t$, we further take it as the adaptive upper bound for $\alpha_t/\sqrt{V_t}$ to eliminate extremely-large learning rates:

$$\eta_t = \min(\hat{\eta}_t, \alpha_t/\sqrt{V_t}). \tag{3}$$

where η_t is the final learning rate obtained by the bounding operation. Intuitively, this can be seen as clipping the learning rates element-wisely so that the output is constrained by the different current momental values. Our proposed momental bounding strategy is significantly different from previous work such as Adafactor and AdaBound. These methods rely on manually chosen thresholds or bounding functions to truncate learning rates or updates. Compared to AMSGrad, AdaMod replace the absolute maximum of previous v_t values with a running average.

$$\theta_t = \theta_{t-1} - \eta_t \odot m_t, \tag{4}$$

Then, we use η_t and m_t to make a parameter update. This process needs to iterate T steps until an approximate solution is returned.

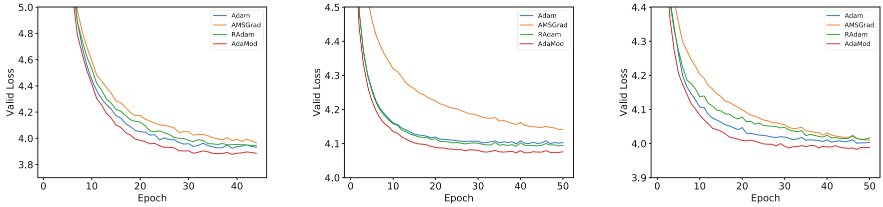
5 Experiments

This section performs a thorough evaluation of AdaMod optimizer on different deep learning tasks against fine-tuned baselines. We refer to several benchmarks for Transformers: **IWSLT'14 De-En/WMT'14 En-De** for neural machine translation, and **CNN-DailyMail** for document summarization. To verify the versatility of AdaMod, we also add image classification on **CIFAR-10/CIFAR-100** [8] with CNN and language modeling on **Penn Treebank** [13] with LSTM. We compare our method with popular optimization algorithms including SGDM, AdaBound, Adam, AMSGrad and RAdam [10]. We classify AdaMod, RAdam, AMSGrad, and Adam into Adam-like algorithms, and the others into SGD-like methods. It is worth noting that except SGDM, the rest belong to adaptive methods. To achieve better performance, we apply decoupled weight decay to all adaptive methods in our experiments on the basis of [11]'s work, and adopt the warmup scheme for all the methods except AdaMod and RAdam when training Transformers. For statistical significance, we conduct each experiment for 5 random trials and report $p - value$ for the significance test. Noting that full hyperparameter tuning details and more experiments results (running Pre-Norm Transformers on IWSLT'14, CNN on CIFAR-10/CIFAR-100, and LSTM on Penn Treebank) are reported in the supplementary material.

5.1 Neural Machine Translation

Machine translation is one of the most important applications in NLP [21]. To evaluate the effectiveness of AdaMod, we train Transformer-based models on two widely used datasets: IWSLT'14 De-En and WMT'14 En-De.

Our experiments are based on the vanilla Transformers [21] implementation from the *fairseq* open library [14]. Due to the limited size of the IWSLT'14 dataset, we use a relatively small model in training. The size of embeddings and hidden states is set to 512 and the number of heads in multi-head attention is set to 4. For WMT'14, we train the transformer base version and the big version respectively. Both the two models consist of a 6-layer encoder and a 6-layer decoder. The size of the embedding is set to 512 for the base model and 1024 for the big. We set $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 1e - 9$. We use a linear warmup for Adam in the first 4000 updates. For IWSLT'14, we run training on



(a) Transformer-Small (b) Transformer-Base (c) Transformer-Big

Fig. 2. Validation loss for Transformer-based model. For (a) is trained on IWSLT'14 De-En, (b) and (c) on WMT'14 En-De. AdaMod without warmup shows both faster convergence and strong final performance compared with Adam with warmup.

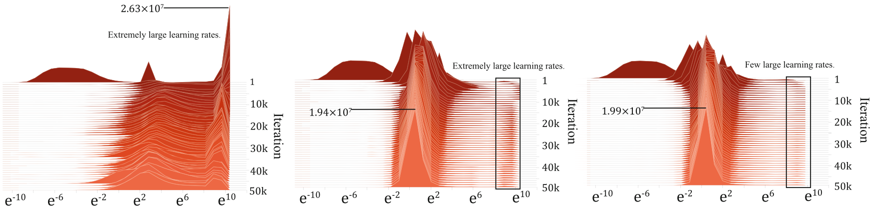
Table 1. BLEU score on Neural Machine Translation. We train the small Transformer on IWSLT'14 De-En, the base and the big model on WMT'14 En-De. Report for $Median(Mean \pm Std)$.

Method	Trans-Small	Trans-Base	Trans-Big
SGDM	29.52 (29.53 ± 0.26)	23.33 (23.35 ± 0.22)	24.47 (24.57 ± 0.24)
AdaBound	34.28 (34.32 ± 0.18)	27.02 (27.05 ± 0.13)	28.24 (28.25 ± 0.09)
Adam	34.62 (34.58 ± 0.15)	27.11 (27.10 ± 0.13)	28.31 (28.31 ± 0.12)
AMSGrad	33.81 (33.84 ± 0.08)	25.98 (26.01 ± 0.10)	27.46 (27.45 ± 0.05)
RAdam	34.72 (34.71 ± 0.06)	27.15 (27.17 ± 0.06)	28.17 (28.20 ± 0.05)
AdaMod	**34.88** (34.85 ± 0.05)	**27.39** (27.39 ± 0.04)	**28.58** (28.57 ± 0.05)

1 NVIDIA RTX 2080Ti GPU, the maximum tokens per batch is set as 4000, weight decay as 1e-4, and dropout rate as 0.3. As for WMT'14, we conduct training on 4 T V100 GPUs and set the maximum tokens as 8192. Note that γ is set as 0.999 for the base model, and 0.9999 for the other two. To make the experiments more convincing, We also compare AdaMod with Adam on the Pre-Norm Transformers in the supplementary material.

Performance Comparison. We use BLEU [15] as the metric to evaluate the performance and give the results in Table 1. We also report p-value between Adam and AdaMod for the significance test (5.42e−3, 1.92e−3, 1.85e−3, all less than 0.01). As discussed above, Adam relies on the warmup scheme when training Transformers to avoid the non-convergence problem (The same goes for other methods except AdaMod and RAdam). As for AdaMod, it can train Transformers without warmup and achieve higher BLEU scores (around 0.3 gains) on both two datasets. Moreover, valid loss values are shown in Fig. 2. It can be seen that AdaMod converges faster against other methods throughout the training stage.

Learning Rates Comparison. We further compare the learning rates histogram of Transformers on the IWSLT'14 De-En between Adam and AdaMod. As shown



(a) Adam without warmup (b) Adam with warmup (c) AdaMod

Fig. 3. The learning rate comparison of Transformers on the IWSLT'14 De-En. AdaMod subtly restrains extremely-large learning rates throughout the training.

Table 2. F1-ROUGE score for CopyTransformer on CNN-DailyMail. Report for $Median(Mean \pm Std)$.

CNN-DM	ROUGE-1	ROUGE-2	ROUGE-L
SGDM	38.34 (38.30 ± 0.15)	16.53 (16.58 ± 0.16)	35.35 (35.29 ± 0.22)
AdaBound	37.75 (37.72 ± 0.09)	16.07 (16.04 ± 0.11)	34.83 (34.86 ± 0.14)
Adam	39.22 (39.24 ± 0.08)	17.19 (17.18 ± 0.13)	36.38 (36.34 ± 0.11)
AMSGrad	39.04 (39.05 ± 0.02)	16.85 (16.86 ± 0.03)	36.07 (36.06 ± 0.05)
RAdam	39.42 (39.44 ± 0.04)	17.23 (17.25 ± 0.05)	36.44 (36.45 ± 0.03)
AdaMod	**39.51** (39.51 ± 0.03)	**17.37** (17.37 ± 0.03)	**36.80** (36.82 ± 0.06)

in Fig. 3, where the X-axis is the original value in the log scale, and Y-axis is iteration steps and the height stands for frequency. As mentioned above, the histogram of Adam is distorted seriously due to the extremely-large learning rates. This phenomenon has been alleviated after adopting warmup, while there is still a spike of learning rates between e^8 to e^{10}. Although they are not enough to fail to converge, they still hurt the generalization performance. In contrast, AdaMod filters the unexpected large learning rates without warmup. Specifically, in the early stage, AdaMod successfully suppressed the abnormal upward trend in learning rates, and keep them within a reasonable range to balance the gradients across layers

5.2 Document Summarization

We also consider abstractive document summarization task on the CNN-DailyMail corpus, which is a standard news corpus and widely used for text summarization. Following [4]'s work, our experiment is based on a Transformer containing 4 layers in each block, and one of the attention-heads is selected as the copy-distribution. Apart from this, the size of the word embedding is set to 512 and the model shares it between the encoder and decoder. We set $\beta_1 = 0.9$, $\beta_2 = 0.998$, and a linear warmup in the first 8000 updates for all the methods except AdaMod and RAdam. The dropout rate is set as 0.2 and the batch size is

4096. The gradients are accumulated 4 times for each update. For AdaMod, the memory coefficient γ is set as 0.9999. All the models are trained on 2 NVIDIA RTX 2080Ti GPUs.

We evaluate by F1-ROUGE [9], i.e., ROUGE-1, ROUGE-2, ROUGE-L, and show the results in Table 2. Note that all the p-values between Adam and AdaMod are less than 0.01 (1.1e−4, 9.6e−3, 2.3e−5). As we can see from the table, AdaMod brings a significant improvement over Adam, and outperforms the rest methods across all three scores, despite without warmup. That is, not only in terms of Rouge-1 related to content but the fluency of the language is also ameliorated by applying AdaMod. It indicates that our method improves the performance of the model on the basis of solving the non-convergence problem and demonstrates the versatility of AdaMod on different natural language tasks based on Transformers.

6 Analysis

Robustness to Different α. To investigate the robustness of AdaMod, we conduct experiments with the Transformer-small on the IWSLT'14 De-En as in the Sect. 5.1. We test Adam with warmup and AdaMod with different α (i.e. initial step size), which is chosen from multiples of 5e−4. The scores are reported in Table 3. It can be found that Adam is very sensitive to α when training Transformers, and a slight disturbance to α will cause poor convergence (below 10). At this time Adam has to resort to other strategies such as gradient clipping or accumulation to get more stable learning rates. However, AdaMod can still converge to highly similar results in the interval of twice the original step size (34.90 ± 0.1). For example, when α is increased by 2.5 times, it can achieve relatively good performance (above 30).

Table 3. BLEU score of Adam and AdaMod with different α using Transformer-small on IWSLT'14 De-En. "×" denotes divergence.

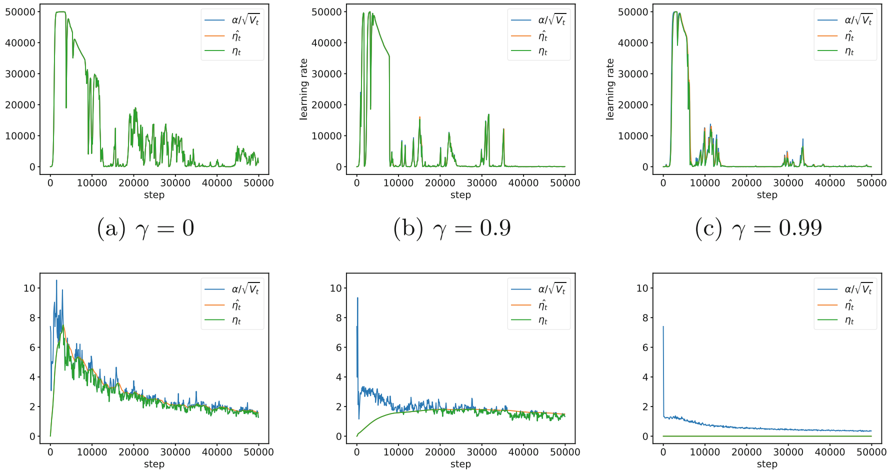
Method	$\alpha = 5e-4$	$\alpha = 7.5e-4$	$\alpha = 1e-3$	$\alpha = 1.25e-3$	$\alpha = 1.5e-3$
Adam	34.62	6.70	1.84	×	×
AdaMod	**34.88**	**34.99**	**34.86**	32.09	×

Training with Small Batches. We also show that AdaMod enables warmup-free, validation-based training even for small batches. Table 4 demonstrates that Adam depends on a large batch size to obtain stable gradients. As the batch size decreases, the performance of the model drops rapidly until Adam fails to converge. In the case of a small batch, the unexpected gradient fluctuations become more drastic, further exacerbating the negative effect of the large learning rates. This gap will become even greater when training complex models. However, AdaMod can converge to similar results (34.85 ± 0.05) by endowing learning rates with long-term memory evne if the batch size has dropped to nearly one-third of the original.

Table 4. BLEU score of Adam and AdaMod with different batches using Transformer-small on IWSLT'14 De-En. "×" denotes divergence.

Method	$bz = 4000$	$bz = 2000$	$bz = 1500$	$bz = 1000$	$bz = 500$
Adam	34.62	33.72	4.43	3.78	×
AdaMod	**34.88**	**34.89**	**34.83**	34.31	3.64

How Momental Bounds Regulate Learning Rates? As discussed above, $\alpha/\sqrt{V_t}$ have a large variance in the early training stage, leading to extremely-large learning rates, which could hamper performance and even cause stability issues.



(a) $\gamma = 0$ (b) $\gamma = 0.9$ (c) $\gamma = 0.99$

(d) $\gamma = 0.999$ (e) $\gamma = 0.9999$ (f) $\gamma = 1$

Fig. 4. How momental bounds regulate learning rates? The learning rates of a parameter were randomly sampled from the encoder of the Transformer. As γ increases, the growth trend of $\alpha/\sqrt{V_t}$ is suppressed by momental bound $\hat{\eta}_t$, and a new stable learning rates η_t is returned. Noting that when $\gamma = 0$, AdaMod degenerates to Adam.

By computing the momentum $\hat{\eta}_t$, historical statistics of learning rates could be fully considered. Employing it as the momental bound could adaptively suppress the growth trend of learning rates. To visualize this process, we supplement an analysis experiment on IWSLT'14. Specifically, we randomly sample a parameter from the self-attention layer to observe its learning rate, and the phenomena are shown in Fig. 4. It can be seen that when historical statistics are not considered (i.e. $\gamma = 0$) or statistics insufficient (i.e. $\gamma = 0.9, 0.99$), the effect of momental bounds is not obvious. The parameter's learning rate surges to 50000 in the early term and collapses dramatically in the later period. However as γ increases, the growth trend of $\alpha/\sqrt{V_t}$ is suppressed by momental bound (i.e. $\gamma = 0.999, 0.9999$), and a new stable learning rates η_t is returned, eliminating large rates from the root, which verifies our motivation. When $\gamma \to 1$, η_t becomes more stable and reliable, which can be approximated as a tailor-made learning rate applied to each parameter of the model, rather than scales all the parameters uniformly like SGD. It is likely that AdaMod combines the advantages of both types of adaptive and non-adaptive methods. *We recommend a γ in $\{0.999, 0.9999\}$ as preferred for its usually behaving a good performance across most models in practice.*

7 Conclusion

In this paper, we illustrate that popular adaptive algorithms fail to converge when training Transformers due to the large learning rates. For that, we design a concise strategy to constrain the spikes to avoid the non-convergence issue. Our proposed algorithm, AdaMod, exerts momental bounds on a per-parameter basis to prevent them from becoming undesirably larger than what the historical statistics suggest, hence getting more stable and reliable learning rates. Empirical results demonstrate our method gains steady performance improvements to Transformers across different applications.

References

1. Chen, M.X., et al.: The best of both worlds: Combining recent advances in neural machine translation. In: ACL (1), pp. 76–86. Association for Computational Linguistics (2018)
2. Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q.V., Salakhutdinov, R.: Transformer-xl: attentive language models beyond a fixed-length context. In: ACL (1), pp. 2978–2988. Association for Computational Linguistics (2019)
3. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
4. Gehrmann, S., Deng, Y., Rush, A.M.: Bottom-up abstractive summarization. In: EMNLP, pp. 4098–4109. Association for Computational Linguistics (2018)
5. Gotmare, A., Keskar, N.S., Xiong, C., Socher, R.: A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In: ICLR (Poster). OpenReview.net (2019)
6. Huang, H., Wang, C., Dong, B.: Nostalgic adam: weighting more of the past gradients when designing the adaptive learning rate. In: IJCAI, pp. 2556–2562. ijcai.org (2019)

7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
8. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep, Citeseer (2009)
9. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches out, pp. 74–81 (2004)
10. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265 (2019)
11. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101 (2017)
12. Luo, L., Xiong, Y., Liu, Y., Sun, X.: Adaptive gradient methods with dynamic bound of learning rate. In: ICLR (Poster). OpenReview.net (2019)
13. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the penn treebank. Comput. Linguist. **19**(2), 313–330 (1993)
14. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: NAACL-HLT (Demonstrations), pp. 48–53. Association for Computational Linguistics (2019)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
16. Popel, M., Bojar, O.: Training tips for the transformer model. Prague Bull. Math. Linguistics **110**, 43–70 (2018)
17. Reddi, S.J., Kale, S., Kumar, S.: On the convergence of adam and beyond. In: ICLR. OpenReview.net (2018)
18. Robbins, H., Monro, S.: A stochastic approximation method. The annals of mathematical statistics, pp. 400–407 (1951)
19. Shazeer, N., Stern, M.: Adafactor: adaptive learning rates with sublinear memory cost. In: ICML. Proceedings of Machine Learning Research, vol. 80, pp. 4603–4611. PMLR (2018)
20. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks Mach. Learn. **4**(2), 26–31 (2012)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
22. Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D.F., Chao, L.S.: Learning deep transformer models for machine translation. In: ACL (1), pp. 1810–1822. Association for Computational Linguistics (2019)
23. Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B.: The marginal value of adaptive gradient methods in machine learning. In: Advances in Neural Information Processing Systems, pp. 4148–4158 (2017)
24. Xiong, R., et al.: On layer normalization in the transformer architecture. CoRR abs/2002.04745 (2020)
25. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
26. Zhou, Z., Zhang, Q., Lu, G., Wang, H., Zhang, W., Yu, Y.: Adashift: decorrelation and convergence of adaptive learning rate methods. In: ICLR (Poster). OpenReview.net (2019)

Oral: Machine Translation and Multilinguality

Towards Making the Most of LLM for Translation Quality Estimation

Hui Huang[1], Shuangzhi Wu[2], Xinnian Liang[3], Bing Wang[3], Yanrui Shi[4],
Peihao Wu[2], Muyun Yang[1](✉), and Tiejun Zhao[1]

[1] Faculty of Computing, Harbin Institute of Technology, Harbin, China
huanghui@stu.hit.edu.cn, {yangmuyun,tjzhao}@hit.edu.cn
[2] ByteDance AI Lab, Beijing, China
{wufurui,wupeihao}@bytedance.com
[3] State Key Lab of Software Development Environment,
Beihang University, Beijing, China
{xnliang,wangbing}@buaa.edu.cn
[4] Global Tone Communication Technology Co., Ltd., Beijing, China
shiyanrui@gtcom.com.cn

Abstract. Machine Translation Quality Estimation (QE) aims to evaluate the quality of machine translation without relying on references. Recently, Large-scale Language Model (LLM) has made major breakthroughs, and has shown excellent zero-shot ability on various natural language processing tasks. However, its application on QE is non-trivial and has not yet been explored. In this work, we aim to exploit the translation estimation ability of LLM, and propose an unsupervised QE framework via exploring the useful information that can be extracted from the LLM. We firstly formulate QE in a machine translation template, and derive the sequence-level probabilities as the translation estimation result. Moreover, we exploit the uncertainty of LLM as another QE evidence, by randomize the LLM with different demonstrations and prompts, and obtain the variance. We evaluate our method on WMT'22 QE data, and achieve high correlation with human judgments of quality, rivalling state-of-the-art supervised QE models. We also provide in-detailed analysis on the ability of LLM on QE task.

Keywords: Large Language Model · Translation Quality Estimation · Machine Translation

1 Introduction

Quality Estimation (QE) aims to predict the quality of machine translation automatically in the absence of reference translations. As state-of-the-art QE models mainly fall into the medium-scale pre-trained model-based paradigm (such as multilingual BERT [6], XLM-Roberta [5]), thousands of annotated samples is

H. Huang—Contribution during internship at ByteDance Inc.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 375–386, 2023.
https://doi.org/10.1007/978-3-031-44693-1_30

required for task-specific fine-tuning, restricting its application. Besides, it has been demonstrated that the supervised QE model is biased and could not generalize well on different domains and languages [3,24].

Recently, large-scale language models pretrained on massive unlabeled corpora, such as GPT3.5 and ChatGPT, have shown impressive abilities on various natural language processing tasks. With the help of task-specific prompting, LLM can formulate various tasks as language generation in a zero-shot manner [1,12,14,20]. Hence, we want to propose a question: Is it feasible to prompt LLMs to perform zero-shot QE under a unified framework? It is challenging since LLM is trained to predict the next token in a natural sentence, while QE needs to assign a continuous score for each sequence.

In this work, we aim at filling this gap by exploiting the translation quality estimation ability of LLMs. Instead of prompting the LLM to perform text evaluation, we propose to prompt the LLM to perform machine translation, and derive the estimation result by exploring the useful information inside the LLM. Firstly, we fill the sample in the translation template, and derive the log-likelihood of the target sequence which reveals the confidence of LLM on the translation result, and can be directly utilized as the estimation result. Moreover, we manage to obtain the model uncertainty as another estimation evidence. We quantify the uncertainty by forward-passing the same sample with different prompts and demonstrations, and obtain the variance of sequence probabilities. If the translation result is fluent and adequate, then the LLM should be robust and the uncertainty should be low, and vise versa. By combining the confidence estimation and uncertainty quantification, we manage to estimation the quality in an unsupervised manner without task-specific fine-tuning.

We conduct experiments on the WMT'22 QE Evaluation data [28], including three directions, namely English-German, Chinese-English and English-Russian. Experiments show that our approach can achieve high correlation with human judgments of quality, rivalling state-of-the-art supervised methods.

2 Background

2.1 Quality Estimation

In contrast to the automatic MT evaluation metrics (such as BLEU [18]) which is good at system level, QE is usually conducted in either sentence-level or word-level [10]. In this work, we mainly concentrate on sentence-level QE, where the translation quality is measured with different schemes, and the QE model is supposed to provide a quality score for each MT output with its source alongside.

Quality Estimation was proposed as early as in 2004 [4]. After the emergence of BERT [6], Pre-Trained Models (PTMs) soon become popular in the area of QE [2,8]. By pretraining on massive multilingual text, PTMs can be adapted to quality estimation task without complicated architecture engineering and achieve remarkable results. However, annotated data with human-crafted quality label is still required for fine-tuning, which is not easily accessible. Besides, it has been revealed that the supervised QE models are relying on spurious clues (such

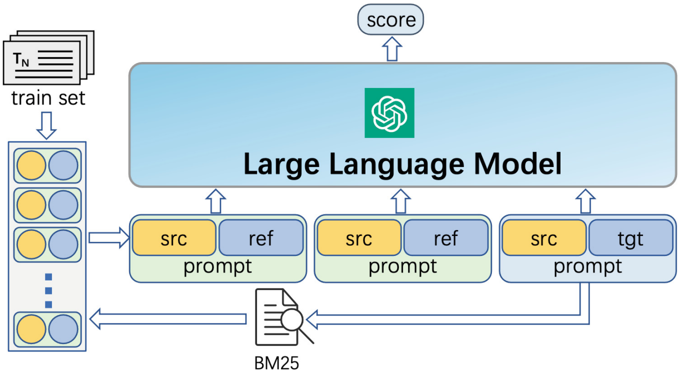


Fig. 1. The architecture of our QE method.

as monolingual fluency) to make prediction [3,24], and cannot generalized well among different domains and languages.

2.2 Large-Scale Language Model

With the advent of ChatGPT, the field of natural language processing has undergone a revolution. As an advanced large-scale language model, ChatGPT has brought about unprecedented semantic understanding and response generation capabilities, and exhibit excellent performance in various traditional natural language processing tasks [1,12,14,20].

Concurrent with our work, Lu [16] and Kocmi [11] also explore the application of LLM on translation evaluation. However, both of them formulate the prompt as an evaluation task and obtain the result via generation, ignoring the information inside the LLM, and their main contribution focus on the designing of prompts. Besides, they only explore the reference-based quality evaluation, while more scalable reference-free QE remains unexplored.

3 Approach

3.1 QE as Confidence Estimation

A straightforward method to apply LLM on QE is to prompt the LLM to evaluate the given translation and generate a score, or on other words, designing the prompt as "evaluation-style" as follows:

```
   Score the following translation from {source_lang} to
{target_lang} on a continuous scale from 0 to 100, where score of
zero means ‘‘no meaning preserved’’ and score of one hundred means
‘‘perfect meaning and grammar’’.

                  {source_lang} source: "{source_seq}"
```

```
{target_lang} translation: "{target_seq}"
```

In this way, the estimation result is obtained in the form of tokens by the decoder. However, to further explore the potential of LLM on QE, we choose to formulate the prompt in "translation style" as follows, and explore the confidence information of the LLM as the estimation result:

```
Please translate {source_seq} into {target_lang}: {target_seq}
```

As shown in Fig. 1, the to-be-evaluated source and target sequences are fed to the LLM with the above prompt, and instead of generating new tokens, we derive the probabilities of target sequence as follows:

$$\mathrm{SP}(h|s) = \sum_{t=m}^{m+n} w_t \log p(h_t|h_{<t}, \mathrm{T}(s,t), \theta)$$

where h_m to h_{m+n} is the span of target tokens, and w_t is the weight of the token at position t. In our work, we treat each token equally. $T(\cdot)$ is the prompt template, and θ is the model parameters of LLM. Intuitively, if the translation result is of high quality, then it will be assigned with a higher probability by the LLM, and vice versa. Therefore, the derived probabilities can be directly used as the estimation result.

This is inspired by the confidence estimation of neural machine translation model, and can be performed in an unsupervised manner [7]. Notably, our method does not require the accessibility of the translation model, therefore is more scalable. Also, as can be seen, compared with evaluation-style prompt, translation-style prompt does not require detailedly illustrating the evaluation aspects and scoring scales and is much simpler.

Previous research shows that a few annotated samples can improve the performance of LLM via In-context Learning [16,17]. Therefore, we extend the prompt with a few annotated samples as demonstration. As shown in Fig. 1, parallel sentence pairs are selected from the training set, and filled in the same translation-style prompt, and then concatenated as prefix to be fed together.

More specifically, we resort to BM25 [23], a similarity-based measurement, to elicit similar samples. Overlapping n-grams between the source and the retrieved sentences ensures informativeness as the target associated with the retrieved sentence is likely to include partial translations of the source. We compute a recall-based (R) n-gram overlap score using the following equation:

$$\mathrm{R_n} = \frac{\sum_{ngram \in S \cap Q} \mathrm{Count}_{matched}(ngram)}{\sum_{ngram \in S} \mathrm{Count}_S(ngram)}$$

where S and Q are the source part of the example and to-be-evaluated sample, respectively. The examples with the maximum scores are then added to the set of selected demonstrations.

Table 1. Different Templates for QE

Translate {src_seq} into {tgt_lang}: {tgt_seq}
Please translate {src_seq} into {tgt_lang}: {tgt_seq}
Help me to translate {src_seq} into {tgt_lang}: {tgt_seq}
Translate {src_seq} from {src_lang} into {tgt_lang}
Please translate {src_seq} from {src_lang} into {tgt_lang}: {tgt_seq}
Help me to translate {src_seq} from {src_lang} into {tgt_lang}: {tgt_seq}
{src_lang}: {src_seq}; {tgt_lang}: {tgt_seq}
{src_lang} source: {src_seq}; {tgt_lang} translation: {tgt_seq}
The {tgt_lang} translation of {src_lang} is: {src_seq}

Notice while In-context Learning can also be applied to evaluation-style template, their demonstration requires samples with score annotation, while we only need parallel sentence pairs which is more easily accessible.

3.2 QE as Uncertainty Quantification

Uncertainty quantification, which quantifies how confident a mapping is with respect to different inputs, has made significant progress due to the recent advances in Bayesian deep learning [13,27]. In this work, we aim to calculate model uncertainty [27], which measures whether a model can best describe the data distribution. A relevant method of approximation, Monte Carlo dropout [9], is usually considered to be useful. It enables random dropout on neural networks during inference to obtain measures of uncertainty. Output sampled across stochastic forward-passes with sampled model parameters θ can be different. Intuitively, if the translation result is a high-quality output with small uncertainty, the sampled outputs should be similar and the diversity among them should be low [26].

However, as most of LLMs are only accessible via API, we are unable to perform dropout during inference. Therefore, in this work, we manage to obtain uncertainty quantification by randomized input. As Monte Carlo Dropout approaches can be regarded as a robustness test with randomness in the model, due to its validity in Fomicheva [7] and Wang [25], it is rational to believe that a similar way with randomized input will perform comparably.

More specifically, we design different prompt templates and prepare different demonstrations, and then feed the same sample to the LLM multiple times with different inputs. Based on that, the expectation of sentence-level translation probability can be approximated by:

$$\text{SP-Exp} = \frac{1}{N}\sum_{n=1}^{N} \text{SP}_{T^n}$$

where SP denotes the sentence level probabilities derived in Sect. 3.1, and N is the forward-pass number.

Moreover, the variance of sentence-level translation probability can be approximated by:

$$\text{SP-Var} = \mathbb{E}[\text{SP}^2_{T^n}] - \text{SP-Exp}^2$$

which is also referred to as model uncertainty.

Finally, we look at a combination of the two:

$$\text{SP-Combo} = Norm(\text{SP-Exp}) + Norm(\text{SP-Var})$$

$$Norm(x_i) = \frac{Max(x_i) - x_i}{Max(x_i) - Min(x_i)}$$

where $Norm$ is to normalize the logit scale into 0 to 1. With the combination of expectation and variance, we take both confidence estimation and uncertainty quantification into consideration, exploiting the useful information accessible in the LLM. While previous works achieve this combination by more complicated operations [26], we found a simple addition can achieve the best performance.

To construct different inputs for stochastic forward-pass, we randomly choose the demonstration from the top-n retrieved samples, and randomly choose prompt from the following templates listed in Table 1.

4 Experiments

4.1 Set-Up

We mainly conduct our experiments on WMT'22 QE Evaluation Dataset [28], including three language directions: English-German, English-Russian, Chinese-English. This dataset consists of triplets with (src_seq, tgt_seq, score), where the score is annotated based on the multi-dimensional quality metrics (MQM) [15] (Table 2).

Table 2. Data Statistics of WMT'22 QE Evaluation Task

Language Pair	train	dev	test
English-German	28909	1005	511
Chinese-English	35327	1019	505
English-Russian	15628	1005	511

We mainly compare our method with several state-of-the-art supervised methods for QE, including TransQuest [21] and COMET [22]. For TransQuest, we use the MonoTransQuest architecture with the default config. For COMET, we use the referenceless architecture with the default referenceless config. Both methods are based on pre-trained model, and we implement their methods on XLM-RoBERTa-base [5][1].

We also compare our method with the submission results taken from WMT'22 [28]. The baseline submission is a fine-tuned model with the XLM-RoBERTa-large [5] as encoder, while the best submissions are ensemble models with various data augmentation techniques, therefore should serve as the upper-bound.

We utilize text-davinci-003[2] as our backbone model. For our unsupervised method, we construct demonstrations from the train set, and directly infer on test set. For the supervised methods, we fine-tune the model on the train set and pick up the best checkpoint on dev set, to be inferred on the test set. We set the random forward-pass number N as 10 in all experiments.

Spearman's Rank Correlation Coefficient (SRCC) between the prediction and the human annotation is taken as the major metric according to WMT'22.

4.2 Results

Table 3. Experiment results on WMT'22 QE test set of different QE methods.

Model	English-German		Chinese-English		English-Russian	
	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
WMT22-best	0.6350	0.6298	0.3480	0.2850	0.5190	0.4076
WMT22-baseline	0.4550	0.4235	0.1640	0.0049	0.3330	0.2791
TransQuest [3]	0.4146	0.4713	0.1758	0.0394	0.2622	0.2606
COMET [22]	0.4120	0.3985	0.1905	0.0059	0.2733	0.1806
SP	0.4534	0.3350	0.2844	0.0294	0.3613	0.2181
SP-Exp	0.4788	0.3490	0.2816	0.0368	0.3821	0.2215
SP-Var	0.4335	0.3785	0.2315	0.1476	0.3455	0.2098
SP-Combo	**0.5116**	0.4110	**0.3059**	0.0791	**0.3920**	0.2393

As can be seen in Table 3, with only confidence estimation (SP), we are able to surpass the fine-tuning based methods by a large margin, and the enhanced version with uncertainty quantification (SP-Combo) can introduce further improvement. While the variance-based method (SP-Var) falls behind the expectation (SP-Exp), it can provide estimation evidence from a different perspective, and the combination of both (SP-Combo) can achieve the best performance. It has

[1] It should be noticed that we did not use any released checkpoint provided by these quality estimation systems, since we want to make a fair comparison in the same data setting, and it is not clear what data augmentation technique is used in training their checkpoints.

[2] https://platform.openai.com/docs/models/gpt-3-5.

to be addressed that our method is unsupervised and can be implemented in a zero-shot manner, with only a few parallel sentence pairs as demonstrations, while the supervised methods requires annotation data and task-specific fine-tuning. With the excellent ability of LLM, we are able to unify QE in different scenarios in a universal framework.

5 Analysis

5.1 The Influence of Demonstration

Table 4. Results on WMT22 QE test set with different demonstrations.

Method	English-German		Chinese-English		English-Russian	
	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
0 demo	0.4132	0.2938	0.1896	0.0210	0.2933	0.1631
1 random demo	0.4484	0.3234	0.2769	0.0302	0.3547	0.2103
1 BM25 demo	0.4534	0.3350	0.2844	0.0294	0.3613	0.2181
10 BM25 demo	0.4788	0.3590	0.2816	0.0368	0.3821	0.2215
SP-Combo	0.5116	0.4110	0.3059	0.0791	0.3920	0.2393

According to previous research [17], a few annotated samples can be of great help to improve the LLM's ability on specific tasks. Since we are using different demonstrations for the same sample to implement uncertainty quantification, it is natural to doubt that does the improvement come from the increase of demonstration numbers instead of the uncertainty quantification?

To answer this question, we perform experiments by concatenating different demonstrations in a single sequence and perform only one single forward pass, and derive the SP as the estimation result. We also try other methods to prepare demonstrations, including randomly choose demonstration every time for each sample, or remove the demonstrations.

As can be seen in Table 4, compared with no demonstration provided, the introduction of just one demonstration could introduce notable improvement, verifying the effectiveness of In-context Learning. And the BM25-based selection can obtain more similar source segments and provide more reference information, therefore outperforms randomly selecting demonstrations. On the other hand, concatenating all demonstrations in a single pass is inferior to feeding them in ten forward-passes and obtain the SP-Combo, verifying the effectiveness of the information obtained through uncertainty quantification.

5.2 QE as Text Generation

As discussed in Sect. 3.1, QE requires the model to assign a score for each sample from a continuous scale. Therefore, we think it is not suitable to be formulated as a generation task. To verify this, we apply LLM to QE with translation-style prompt as described in Sect. 3.1, and derive the estimation result via text generation. We implement this method on two LLMs, including GPT3.5(text-davinci-003) and ChatGPT(gpt-3.5-turbo).

Table 5. Results on WMT22 QE Test set with different task formulations.

Model	Method	English-German		Chinese-English		English-Russian	
		Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
GPT3.5	SP	0.4534	0.3350	**0.2844**	0.0294	**0.3613**	0.2181
GPT3.5	generation	0.3605	0.2421	0.2254	0.0260	0.2837	0.1545
ChatGPT	generation	**0.4568**	0.4993	0.2637	0.3121	0.3200	0.2865

As can be seen in Table 5, when applying GPT3.5 on QE, the results obtained via text generation is much lower than confidence estimation, suggesting that QE is not suitable to be formulated as a generation task. However, while applying ChatGPT with generation-style prompt, it can achieve excellent performance, even comparable with GPT3.5-based confidence estimation method. We believe ChatGPT can produce better results with confidence estimation and uncertainty quantification, but the OpenAI API does not provide the option to expose log probability of ChatGPT, therefore we leave this as the future work.

5.3 Self-generated Demonstration

As discussed in previous research, LLMs such as GPT3.5 can achieve notable performance on machine translation [19, 29]. Therefore, it is natural to think that can we use LLM to create the translation as an extra reference? To this end, we prompt the LLM to translate the source segment of each sample, and obtain the result as the demonstration. We also use the golden reference as demonstration, to illustrate the effectiveness of the self-generated reference.

Since there is no reference provided in the test set, we randomly sample 500 samples from the train set, and compare different demonstrations by filling both references into the following prompts:

```
Please translate {source_seq} into {target_lang}: {reference}
Please translate {source_seg} into {target_lang}: {target_seq}
```

As can be seen from the Table 6, the human-crafted golden reference can introduce massive improvement, suggesting that reference-based method generally outperforms reference-free method on translation evaluation. Reference should always be included in the evaluation scheme if provided.

However, to our surprise, the self-generated demonstration would lead to severe performance degradation, even underperforms the similarity-based

Table 6. Results on our randomly sampled training set with different demonstrations.

Demonstration	English-German		Chinese-English		English-Russian	
	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
self-generated ref	0.3206	0.1986	0.5174	0.4812	0.3485	0.2956
golden ref	0.3657	0.2198	**0.5194**	0.4708	**0.4403**	0.4122
1 BM25 demo	**0.3742**	0.2022	0.5071	0.4357	0.3708	0.2984
0 demo	0.3439	0.1808	0.4989	0.4300	0.3491	0.2859

demonstrations. We think this is because the generated reference is unavoidably noisy and would bias the evaluation process. We calculate the BLEU Score of the original translation and generated reference against the golden reference with sacrebleu[3], and reveals that the reference generated by ChatGPT is actually worse than the original translation, as shown in Table 7. A bad demonstration is even worse than no demonstration.

Table 7. The BLEU Score between the original translation and the self-generated reference against the golden reference.

Language Pair	BLEU Score	
	mt	self-ref
English-German	31.81	29.90↓
Chinese-English	33.24	30.19↓
English-Russian	31.13	21.46↓

6 Conclusion

In this work, we exploit the useful information in LLM and propose an unsupervised framework for QE. We firstly formulate QE in a machine translation prompt, and derive the sequence-level probabilities as the estimation result for translation. Moreover, we exploit the uncertainty of LLM as another QE feature, by randomize the LLM with different demonstrations and prompts. Experiment on WMT'22 QE test set show we are able to surpass state-of-the-art supervised QE methods, demonstrating its effectiveness.

As LLM-based paradigm is taking the NLP community by storm, it deserves our belief that text evaluation would also shift to the new paradigm. In the future, we would extend our method to other text generation tasks, such as summarize evaluation, dialogue evaluation, to create a unified picture for text generation evaluation with the help of LLM.

[3] https://github.com/mjpost/sacrebleu.

Acknowledgements. This work is supported by National Key R&D Program of China (2020AAA0108000, 2020AAA0108005), National Natural Science Foundation of China (62276077, U1908216), Key R&D Program of Yunnan (202203AA080004) and Shenzhen College Stability Support Plan (No. GXWD20220811170358002).

References

1. Bang, Y., et al.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity (2023)
2. Barrault, L., et al.: Findings of the 2020 conference on machine translation (WMT20). In: Proceedings of the Fifth Conference on Machine Translation, pp. 1–55. Association for Computational Linguistics, November 2020
3. Behnke, H., Fomicheva, M., Specia, L.: Bias mitigation in machine translation quality estimation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1475–1487. Association for Computational Linguistics, May 2022
4. Blatz, J., et al.: Confidence estimation for machine translation. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 315–321. COLING, Aug 23-Aug 27 2004
5. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Association for Computational Linguistics (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (Jun 2019)
7. Fomicheva, M., et al.: Unsupervised quality estimation for neural machine translation. Trans. Assoc. Comput. Linguistics **8**, 539–555 (2020)
8. Fonseca, E., Yankovskaya, L., Martins, A.F.T., Fishel, M., Federmann, C.: Findings of the WMT 2019 shared tasks on quality estimation. In: Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), pp. 1–10. Association for Computational Linguistics, August 2019
9. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059. PMLR (2016)
10. Han, L., Jones, G.J., Smeaton, A.F.: Translation quality assessment: a brief survey on manual and automatic methods. arXiv preprint arXiv:2105.03311 (2021)
11. Kocmi, T., Federmann, C.: Large language models are state-of-the-art evaluators of translation quality (2023)
12. Kocoń, J., et al.: Chatgpt: Jack of all trades, master of none (2023)
13. Lee, G., Hou, B., Mandalika, A., Lee, J., Choudhury, S., Srinivasa, S.S.: Bayesian policy optimization for model uncertainty (2019)
14. Liang, P., et al.: Holistic evaluation of language models (2022)
15. Lommel, A., Uszkoreit, H., Burchardt, A.: Multidimensional quality metrics (MQM): a framework for declaring and describing translation quality metrics. Tradumàtica **12**, 0455–0463 (2014)
16. Lu, Q., Qiu, B., Ding, L., Xie, L., Tao, D.: Error analysis prompting enables human-like translation evaluation in large language models: a case study on chatgpt (2023)

17. Min, S., et al.: Rethinking the role of demonstrations: what makes in-context learning work? In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 11048–11064. Association for Computational Linguistics, December 2022
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, July 2002. https://doi.org/10.3115/1073083.1073135
19. Peng, K., et al.: Towards making the most of chatgpt for machine translation. arXiv preprint arXiv:2303.13780 (2023)
20. Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., Yang, D.: Is chatgpt a general-purpose natural language processing task solver? (2023)
21. Ranasinghe, T., Orasan, C., Mitkov, R.: TransQuest: translation quality estimation with cross-lingual transformers. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5070–5081. International Committee on Computational Linguistics (Dec 2020)
22. Rei, R., Stewart, C., Farinha, A.C., Lavie, A.: COMET: a neural framework for MT evaluation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2685–2702. Association for Computational Linguistics, November 2020
23. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retrieval **3**(4), 333–389 (2009)
24. Sun, S., Guzmán, F., Specia, L.: Are we estimating or guesstimating translation quality? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6262–6267. Association for Computational Linguistics, July 2020
25. Wang, K., Shi, Y., Wang, J., Zhang, Y., Zhao, Y., Zheng, X.: Beyond glass-box features: uncertainty quantification enhanced quality estimation for neural machine translation. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 4687–4698. Association for Computational Linguistics, Novenber 2021
26. Wang, S., Liu, Y., Wang, C., Luan, H., Sun, M.: Improving back-translation with uncertainty-based confidence estimation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 791–802. Association for Computational Linguistics (Nov 2019)
27. Xiao, Y., Wang, W.Y.: Quantifying uncertainties in natural language processing tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7322–7329 (2019)
28. Zerva, C., et al.: Findings of the WMT 2022 shared task on quality estimation. In: Proceedings of the Seventh Conference on Machine Translation (WMT), pp. 69–99. Association for Computational Linguistics, December 2022
29. Zhang, B., Haddow, B., Birch, A.: Prompting large language model for machine translation: a case study. arXiv preprint arXiv:2301.07069 (2023)

Towards Better Translations from Classical to Modern Chinese: A New Dataset and a New Method

Zongyuan Jiang, Jiapeng Wang, Jiahuan Cao, Xue Gao, and Lianwen Jin(✉)

South China University of Technology, Guangzhou, China
{eejiangzongyuan,eejpwang}@mail.scut.edu.cn
{xuegao,eelwjin}@scut.edu.cn

Abstract. Classical Chinese (Ancient Chinese) is the written language that was used in ancient China and has been an important carrier of Chinese culture for thousands of years. Numerous ideas of modern disciplines have been influenced or derived from it, including mathematics, medicine, engineering, etc., which demonstrated the necessity for us to understand, inherit and disseminate it. Consequently, there is an urgent need to develop neural machine translation to facilitate the comprehension of classical Chinese sentences. In this paper, we introduce a high-quality and comprehensive dataset called C2MChn, consisting of about 615K sentence pairs for the translation between classical and modern Chinese. To the best of our knowledge, this is the first dataset covering a wide range of domains including history books, Buddhist classics, Confucian classics, etc. Furthermore, through the analysis of classical and modern Chinese, we have proposed a simple yet effective method, named Syntax-Semantics Awareness Transformer (SSAT). It's capable of leveraging both syntactic and semantic information which are indispensable for better translating classical Chinese. Experiments show that our model can achieve better BLEU scores than several state-of-the-art methods as well as two general translation engines including Microsoft and Baidu APIs. The dataset and related resources will be released at: https://github.com/Zongyuan-Jiang/C2MChn.

Keywords: Classical-Modern Chinese Dataset · Syntax-Semantics Awareness · Neural Machine Translation

1 Introduction

Neural machine translation has experienced a remarkable breakthrough, thanks to the rapid advancement of sequence-to-sequence models [3,9,20,21]. It makes it possible to translate an increasing number of language pairs well. However, there still exist some language pair translation issues that have not been well resolved. On the one hand, several languages have suffered from a scarcity of resources [1,12]. On the other hand, a few languages have faded out of sight due

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 387–399, 2023.
https://doi.org/10.1007/978-3-031-44693-1_31

to changes in people's daily language habits [15]. One such language that faces both of these issues is Classical Chinese.

Classical Chinese is the symbol of Chinese history and culture. Chinese ancients used this language to pass down their wisdom and experience to descendants. A better understanding of classical Chinese can help us to inherit Chinese culture better and spread it to more people. Furthermore, by delving into the essence of traditional culture, we can extract valuable insights and apply them to modern society for the benefit of all. A prime example of this is traditional Chinese medicine, which has uncovered many effective remedies that are widely used today. Modern Chinese is the spoken and written language commonly used by Chinese people today. It is characterized by its straightforwardness and ease of comprehension. The most important characteristic of classical and modern Chinese, compared to other language pairs, is they share the most common characters. This commonality makes that same words in both languages may have similar meanings. However, there are still some significant differences between them. Firstly, classical Chinese is the written language of ancient China. It pursues simplicity which means that most of the time it is more concise and shorter than modern Chinese. The translation from classical to modern Chinese will be a process of increasing the amount of information which is difficult. Secondly, the words of classical Chinese are mostly monosyllabic while those of modern Chinese are mostly multisyllabic. Thirdly, the polysemous phenomenon of classical Chinese is more common. Even some words of that have little to do with the corresponding modern Chinese words at a glance. Thus, how to choose the correct translation for the word is a bit difficult. Fourthly, classical Chinese has more obscure sentence expressions, such as judgmental sentences, elliptical sentences, passive sentences, and inverted sentences. These differences make translations from classical to modern Chinese quite challenging.

Therefore, in this paper, we aim to obtain a better neural machine translation from classical to modern Chinese. We first propose a classical-modern Chinese translation dataset, named C2MChn, with 615K bilingual sentence pairs. To the best of our knowledge, this is the first high-quality and comprehensive dataset that not only covers traditional history books but also the Buddhist classics, Confucian classics, Taoist classics, and some other domains. It will greatly advance the development of this research. In addition, through the analysis of classical and modern Chinese mentioned above, we can observe that the monosyllabic and polysemous properties make it more difficult to obtain the correct semantic information. Meanwhile, complex and variable sentence structures make syntactic information more important for such translation. It serves as a reminder for us that both semantic and syntactic information are significant for the translation from classical to modern Chinese. Therefore, we propose a simple yet effective translation method called **Syntax-Semantics Awareness Transformer** (SSAT). The model is based on Transformer [21] with two new modules. To obtain syntactic and semantic information, we use the **Syntax-Semantics Extraction** module to fuse the outputs from the encoder layers in two separate paths. The outputs of two paths will be incorporated in the decoder by the **Dual-path Multi-Head Attention** module. Through this model, we have demonstrated that not a single information of syntax and semantics is dispensable in translating classical Chinese. Furthermore, experiments show that our model outperforms some state-of-the-art (SOTA) methods and two general translation engines including Microsoft and Baidu APIs.

The main contributions of our work are summarized as follows:

- We present a high-quality and comprehensive dataset, named C2MChn, for the translation from classical to modern Chinese. To the best of our knowledge, it is the first dataset covering a wide range of domains including history books, Buddhist classics, Confucian classics, Taoist classics, etc.
- We propose a simple yet effective method, named Syntax-Semantics Awareness Transformer (SSAT), which can leverage both syntactic and semantic information in sentences to achieve better translations. Meanwhile, we have demonstrated empirically that both syntactic and semantic information are indispensable in translations from classical to modern Chinese.
- Experiments show that our model outperforms some SOTA methods and general translation engines.

2 Related Work

Neural Machine Translation. With advances in deep learning, neural machine translation is increasingly becoming the mainstream approach for machine translation. The sequence-to-sequence model is the most commonly used neural machine translation model. This model typically consists of an encoder that captures the contextual information of source sentences and a decoder that generates the translation according to the contextual information from the encoder. The sequence-to-sequence model has evolved through three structures: Recurrent Neural Network [3,11,20], Convolutional Neural Network [9] and Transformer [21]. Transformer which is based solely on attention mechanisms has led to a series of breakthroughs in neural machine translation and other tasks. Consequently, numerous variants of the Transformer have emerged in recent years [6,7,14,19]. Most of these works focus on optimizing the structure to address the shortcomings of Transformer itself to obtain a more general model. They tend to ignore the characteristics of the language itself. In contrast, our model is presented based on the analysis of the language structure itself to make it more appropriate for translation from classical to modern Chinese.

Classical Chinese Translation. As for the translation from classical to modern Chinese, it is gradually gaining more attention but still relatively understudied. Liu et al. [16] proposed a Classical-Modern Chinese clause alignment approach that aligned unpaired Classical-Modern Chinese sentences to create a dataset focused solely on history books. On the contrary, our C2MChn dataset is comprehensive, containing more than six domains of books. It is meticulously checked manually, ensuring its high quality and suitability for translation tasks.

Zhang et al. [26] find that hybrid representational unit (the mixture of characters and words) will bring better translations for classical Chinese in statistical machine translation. Zhang et al. [27] used a bidirectional RNN with the attention mechanism and copying mechanism [10] to achieve a better translation. Chang et al. [5] presented a framework that can predict both the translation and its particular era with additional supervised information. Nevertheless, these works did not focus well on the structure of the two languages themselves. Instead, our studies are based on the analysis of the characteristics and differences between classical and modern Chinese.

Table 1. Some samples from our dataset C2MChn.

Domains	Languages	Sentences
History	Classical	越王勾践用子贡之谋，乃率其众以助吴，而重宝以献遗太宰。
	Modern	越王勾践采用子贡的计谋，就率领他的部队来帮助吴国，并将贵重宝物进献给太宰伯。
	Classical	券遍合，起矫命，以责赐诸民，因烧其券，民称万岁。
	Modern	核对完毕后，冯谖起身假传孟尝君的命令，宣布免掉百姓所欠的债务，并当众把契约烧掉，百姓们欢呼万岁。
Buddhism	Classical	“我今求法，为成佛道，后得佛时，当以智慧光明，悟众生结缚黑暗。”
	Modern	“我今舍身追求道法，是为了掌握佛所证悟的道法，掌握了佛所证悟的道法后，我将以智慧之光，照亮众生的心田，使众生悟出事物的道理，从黑暗、愚昧里解脱出来，永获光明。”
	Classical	生六识，出六门，见六尘，如是一十八界，皆从自性起用。
	Modern	产生眼识，耳识、鼻识、舌识、身识和意识这六识，便可从六门中走出，而认识到色、声、香、味、触、法这六尘。如此这样的一十八界，都是从自然的本质里所产生的。
Confucianism	Classical	人伦并处，同求而异道，同欲而异知，生也。
	Target	人类共处，具有共同的需求而满足需求的方法不一样，具有共同的欲望而实现欲望的智慧却不一样，这是人的本性。
	Classical	於乎！吾以王言之，其不出户牖而化天下。
	Modern	唉，我若把成就王业的道理讲给居高位的人听，他们不出门户就可以治理好天下了。
Taoism	Classical	不知周之梦为胡蝶与，胡蝶之梦为周与？
	Modern	不知道是庄周做梦化为了蝴蝶，还是蝴蝶做梦化为了庄周呢？
	Classical	莫知其所终，若之何其无命也？ 莫知其所始，若之何其有命也？
	Modern	天地运化莫测，生死循环不已，我们怎么能推定何时是生命的开始，何时是生命的结束呢？
Agronomy	Classical	“薄田不能粪者，以原蚕矢杂禾种种之，则禾不虫。”
	Modern	“太瘦的地，又没有能力上粪的，所以用多化蚕的蚕粪，和入种子一齐播种；这样，还可以免除虫害。”
	Classical	“当种麦，若天旱无雨泽，则薄渍麦种以酢且故反浆并蚕矢。
	Modern	“该种麦的时候到了，遇到干旱不下雨，地里又没有足够的墒，就用酸浆水浸上蚕粪，稀稀地泡着麦种。
Short	Classical	范增数目项王，举所佩玉以示之者三，项王默然不应。
	Modern	酒会开始后，范增连连地给项羽使眼色，又几次拨弄着所佩玉向项羽示意，但项羽总是默默地不加理睬。
	Classical	凡天下祸篡怨恨可使毋起者，以相爱生也，是以仁者誉之。
	Modern	凡是天下的祸患、掠夺与怨恨可以让它们不发生的，就是因为人们产生了相爱之心，所以仁义的人都赞美它。
Others	Classical	吾虽掩之甚密，文之甚巧，而肺肝早露，终难自欺；
	Modern	我虽然把过失遮盖得十分秘密，掩饰得十分巧妙；但是在神明看来，我的肺肝，早被看透，马脚全露出来了。到最后还是没有办法欺骗自己。
	Classical	夫为弟子后生，其师，必修其言，法其行，力不足知弗及而后已。
	Modern	作为门人弟子，以他们的老师作为榜样，必定学习他的思想，效法他的行为，直到力量不够、智力不足时才会停止。

3 Dataset

3.1 Data Overview

Our comprehensive **C2MChn** dataset consists of approximately 615k sentence pairs. Some samples of it can be seen in Table 1. The content of our dataset C2MChn mainly covers seven domains: **History**, **Buddhism**, **Confucianism**, **Taoism**, **Agronomy**, **Short** and **Others**. **History** includes books that record the history of China. It is a symbol of the rich and diverse history and culture of China. **Buddhism** refers to religious classics belonging to, or associated with, Buddhism and its traditions. Our dataset is the first publicly available data for research on the translation of Buddhist scriptures. It's noteworthy that our dataset includes over two hundred Buddhist scriptures, such as *The Diamond Sutra*, *The Heart Sutra*, *The Surangama Sutra*, etc. It indicates that our dataset could potentially provide unique values for research, which will advance the comprehension of Buddhist philosophy. **Confucianism** comprises books that convey the philosophy of Confucian scholars, represented by Confucius and Mencius. **Taoism** refers to a series of ancient Chinese works that belong to the Taoist tradition. **Agronomy** contains books that mainly focus on various aspects of agriculture and rural life. **Short** includes sentences from some short classical Chinese articles and poems. And **Others** contains sentences that are too few in number to be grouped into one domain. The size of our dataset C2MChn is larger than that of all language pairs in IWSLT17[1] dataset and some language pairs in WMT14[2] dataset that are commonly used in machine translation. Meanwhile, these datasets typically contain one domain such as news or speeches. To further ensure the feasibility of the research, we inspect and select 10k sentence pairs manually from the dataset as the validation set and test set respectively. The remaining 595k sentences will be used for the training set. The detailed statistical results of the dataset can be seen in Table 2.

Table 2. Statistics of our dataset C2MChn. “pairs”, “cla/s” and “mod/s” denote the number of parallel sentence pairs, the average number of word tokens per classical Chinese sentence and the average number of word tokens per modern Chinese sentence, respectively.

Domains	Total			Training			Validation			Test		
	pairs	cla/s	mod/s	pairs	cla/s	mod/s	pairs	cla/s	mod/s	pairs	cla/s	mod/s
History	283600	19.8	28.7	273969	19.8	28.7	4707	20.2	29.4	4924	20.5	29.9
Buddhism	223602	18.2	28.6	216842	18.1	28.5	3509	18.9	29.8	3251	20.4	32.5
Confucianism	49337	18.3	30.5	47655	18.3	30.4	824	18.6	30.7	858	18.9	31.6
Taoism	10433	19.9	32.3	10061	19.9	32.7	187	20.1	32.6	185	19.4	31.4
Agronomy	9748	12.4	18.6	9459	12.3	18.5	162	12.8	19.4	127	14.0	20.8
Short	6820	16.5	30.2	6623	16.5	30.2	99	15.6	29.5	98	17.5	31.3
Others	31183	20.1	32.1	30114	20.1	32.0	512	21.3	33.8	557	21.1	33.2
All	614723	18.9	28.9	594723	18.9	28.8	10000	19.5	29.8	10000	20.3	31.0

3.2 Data Source

The sentences of classical Chinese comprising our dataset were gathered from a variety of publicly available downloadable corpora on the Internet. Some of them have document-level corresponding translations. Thus, we used a private modified dynamic time warping algorithm for preliminary automatic alignment. The obtained sentence pairs were secondary aligned manually for the high-quality of

[1] https://iwslt.org/.

[2] https://www.statmt.org/wmt14/translation-task.html.

the dataset. Nonetheless, there are a few data such as Buddhist scriptures that almost has no corresponding translation on the Internet. Thus, we recruited some Buddhist followers to translate them. To ensure the highest possible quality of our dataset, the sentence pairs have undergone a thorough round of manual verification and revision. It is worth mentioning that this process consumes a significant amount of our time and resources. We believe that it is highly meaningful and will greatly advance the research of this task.

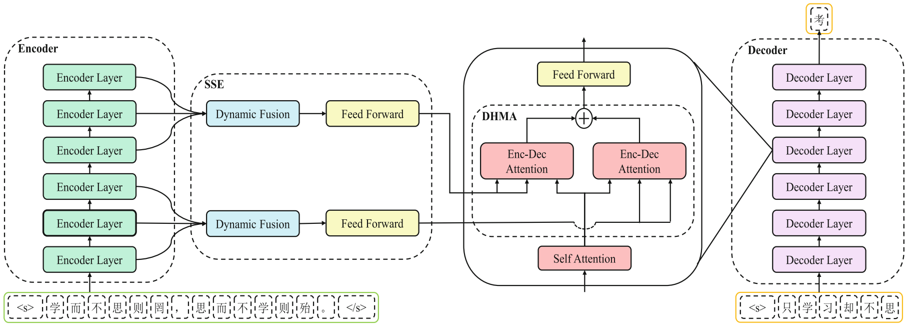


Fig. 1. The overview of the architecture. SSE (Syntax-Semantics Extraction) fuses the syntactic and semantic information from the encoder. DMHA (Dual-path Multi-Head Attention) utilizes them in the decoder to achieve a better translation.

4 Model Architecture

Transformer [21] has demonstrated exceptional performance in machine translation and many other tasks in natural language processing. It follows the paradigm of the encoder-decoder structure. The decoder takes the output from the topmost layer of the encoder as the contextual information. Virtually, outputs from different encoder layers of Transformer represent different information [8,23]. Raganato and Tiedemann [18] have confirmed that lower layers tend to learn more about syntax while higher layers tend to encode more semantics. Meanwhile, as discussed in the previous section, we can observe that both semantic information and syntactic information play a critical role in such translation. Inspired by the above, it is intuitive to leverage both syntactic and semantic information to achieve better translations. Thus, we have proposed a simple yet effective method, named **Syntax-Semantics Awareness Transformer** (SSAT), which can exploit both syntactic and semantic information from sentences. Our model mainly introduces two new modules, the **Syntax-Semantics Extraction** module and the **Dual-path Multi-Head Attention** module. To further use two information from encoder layers, we utilize a Syntax-Semantics Extraction module to dynamically fuse them in two separate paths. The outputs from two paths then will be incorporated in the decoder by a Dual-path Multi-Head Attention module. The overview of our architecture is shown in Fig. 1.

4.1 Syntax-Semantics Extraction

We first divided the Transformer encoder into two pieces. Each piece has three encoder layers following Raganato and Tiedemann [18]. For these two pieces, we introduce two paths to extract syntactic and semantic information which are called syntax path and semantics path respectively. Taking the syntax path as an example, we use the output of the first three encoder layers as the input to the dynamic fusion mechanism (DFM) which is used to dynamically fuse the different outputs from encoder layers. The output $\mathbf{C}_{Sy}$ of DFM can be calculated as:

$$\mathbf{C}_{Sy} = \sum_{l=1}^{L} \alpha^l \mathbf{R}_E^l \tag{1}$$

$$\alpha^l = \frac{\exp(\beta^l)}{\sum_{t=1}^{L} \exp(\beta^t)} \tag{2}$$

$$\beta^l = \frac{1}{L} \sum_{k=1}^{L} (\frac{1}{I} \sum_{i=1}^{I} \mathbf{r}_E^{l,i} \cdot \frac{1}{I} \sum_{i=1}^{I} \mathbf{r}_E^{k,i}) \tag{3}$$

where $\mathbf{R}_E^l$ is the output of l-th Transformer encoder layer and $\mathbf{r}_E^{l,i}$ is the i-th vector of $\mathbf{R}_E^l$. I and L represent the length of the sequence and the number of used layers, respectively. β^l indicates the degree of correlation between the l-th layer and other layers information through cosine similarity. Thus, we can obtain the percentage of each layer in the final syntactic information using a softmax. To further fuse the information from different encoder layers, we add the following process after DFM:

$$\mathbf{R}_{Sy} = \mathrm{FFN}(\mathrm{LN}(\mathbf{C}_{Sy})) + \mathbf{C}_{Sy} \tag{4}$$

where the LN$(\cdot)$ is layer normalization [2] and FFN$(\cdot)$ is feed forward network [21]. We omit the layer normalization and residual connection for the sake of brevity in Fig. 1. For the semantics path, we can get $\mathbf{R}_{Se}$ with the same operations except that the input is replaced by the last three encoder layers.

4.2 Dual-Path Multi-head Attention

To better leverage both syntactic and semantic information from the two paths mentioned above, we use a dual-path multi-head attention. The output of two multi-head attention will be summed by weighting. The process can be expressed in the following equation:

$$\mathbf{H}_{Dy}^l = \mathrm{Attn}(\mathbf{R}_D^l, \mathbf{R}_{Sy}, \mathbf{R}_{Sy}) \tag{5}$$

$$\mathbf{H}_{De}^l = \mathrm{Attn}(\mathbf{R}_D^l, \mathbf{R}_{Se}, \mathbf{R}_{Se}) \tag{6}$$

$$\mathbf{H}_D^l = p \cdot \mathbf{H}_{Dy}^l + (1-p) \cdot \mathbf{H}_{De}^l \tag{7}$$

where Attn(·) is multi-head attention [21]. $\mathbf{R}_D^l$ is the output of self-attention in the l-th decoder layer and p is the proportion of syntactic information we will use. By setting different hyperparameter p, we can experimentally demonstrate the importance of syntactic and semantic information in the translation.

Table 3. Ablation study for each component of our model.

Method	BLEU(△)
Ours	**35.88**
w/o SSE	35.47(−0.41)
w/o DMHA	35.29(−0.59)
w/o Semantics Path	34.36(−1.52)
w/o Syntax Path	35.22(−0.66)

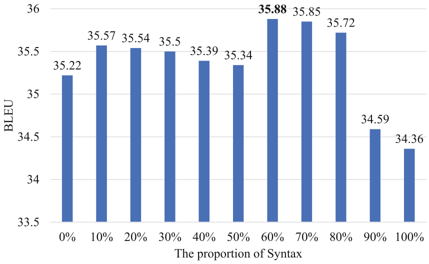


Fig. 2. Changes in BLEU with different proportion of syntax.

5 Experiments

5.1 Implementation Details

We choose Transformer base as our basic model structure. For stable training, we used Pre-LN [22,25] instead of Post-LN. All models are optimized by Adam [13] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We use the inverse square root with a learning rate of $5e^{-4}$ as our learning rate scheduler and the number of warm-ups is set to 4000. Label smoothing $\epsilon_{ls} = 0.1$ is used for regularization. The batch size is set to 8192 tokens per GPU and the max sentence length is limited to 200. For evaluation, we score the translations by `multi-bleu.perl`[3] which is also used by previous works for a fair comparison. And beam search is used as the decoding algorithm with the beam size set to 5.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl.

Table 4. Some translations of different translation systems. Words in red represent translation errors. And words in green mean that the translations are correct.

Source	太子，君嗣也，不可施刑，刑其傅公子虔，黥其师公孙贾。
GT	但太子是国家未来的继承人，不能对他施刑，于是就处罚了太子的太傅公子虔，给太子的太师公孙贾脸上刺了字。
Baidu API	太子，国君的继承人，不可受处罚，惩罚他的老师公子虔，而他的师公孙贾。
Micosoft API	太子，是国君的继承人，不能施以刑罚，就处罚了师傅公子虔，以墨刑处罚了给他传授知识的老师公孙贾。
Ours	太子是国君的继承人，不能对他施刑，就处罚了太子的太傅公子虔，给太子的太师公孙贾脸上刺字。
Source	语曰：“浅不足与测深，愚不足与谋智，坎井之蛙不可与语东海之乐。”
GT	俗语说：“浅陋的人不值得和他谈论深奥的事，愚蠢的人不值得和他谋划智慧的事，废井中的蛤蟆不能和它谈论东海中的快乐。”
Baidu API	俗话说：“浅不足与测深，愚蠢的人不值得与谋智，井里的青蛙不能与对东海的乐趣。”
Microsoft API	俗话说：浅不足与测深，愚不足与智，坎井之蛙不可与之谈东海之乐”
Ours	俗语说：“浅陋的人不值得和他谈论深奥的事，愚蠢的人不值得和他谋划智慧的事，废井中的蛙不能和他谈论东海中的快乐。”
Source	将修小大强弱之义以持慎之，礼节将甚文，璧将甚硕，货赂将甚厚，所以说之者，必将雅文辩慧之君子也。
GT	讲究仁德的人必将小心地遵照小国和大国、强国和弱国之间的道义，礼节将十分完备，会见时赠送的玉器等礼物将十分硕大，进俸的财物十分丰厚，用来游说对方的使者，必定是文雅善辩聪明的君子。
Baidu API	准备大小强弱的意义以谨慎的，礼节会很文，玉器将很大，贿赂将很好，之所以说的，一定要很善辩聪慧的君子啊。
Microsoft API	要想使小、大、强、弱的意义保持谨慎，礼节会十分讲究，玉玉璧将非常肥硕，送的贿赂将要很多，所以，一定要是能雅正修饰、善辩、聪慧的人。
Ours	讲究仁德的人必将小心地遵照小国和大国、强国和弱国之间的道义，礼节将十分完备，赠送的玉器等礼物将十分硕大，进奉的财物十分丰厚，用来游说对方的，必定是文雅善辩聪明的君子。
Source	善心起时，随缘何境，皆有崇重善及轻拒恶义，故惭与愧俱遍善心，所缘无别。
GT	善心生起时，无论认取什么对象，都有崇敬尊重善以及鄙视排斥恶的意义，所以惭与愧都普遍存在于善心之中，二者的认取对象没有差别。
Baidu API	好心情起床时，随着为什么环境，都有高善和恶义轻抵抗，所以惭愧与愧疚一起到处行善之心，为什么没有区别。
Microsoft API	许善心被起用时，随缘何处地，都有崇重善和轻率抵拒恶义，所以羞愧与惭愧都普及善心，无缘无别。
Ours	善心生起时，无论认取什么对象，都有崇重善法和轻视排斥恶法的含义，所以惭与愧都普遍存在于善心之中，所认取的对象没有差别。

5.2 Ablation Study

Effects of Each Component. An ablation study is conducted to explore the effect of each component on the model performance. The results are shown in Table 3. To evaluate the significance of further developing syntactic and semantic information, we replace the Syntax-Semantics Extraction module with a direct summation of outputs of encoder layers. We find that there is a decrease in performance. It reflects the need to further excavate both information. Meanwhile, we directly sum the output of the Syntax-Semantics Extraction module and use it as the input to one multi-head attention instead of the Dual-path Multi-head Attention module. From the results, it suggests that it is much better to explicitly focus on both information separately with two multi-head attention. In addition, we also explore the impact of using either path information on performance. It can be seen that better translations cannot be obtained with only single path information. It indicates that not a single information of syntax and semantics can be dispensable to obtain good translations.

Explorations of the Hyperparameter *p*. To find whether syntactic or semantic information is more important, we conduct an ablation study by adjusting the hyperparameter p of Eq. 7 which stands for the proportion of syntactic information used in the decoder. When p is 0/100%, it means that the process of decoding uses only the semantic/syntactic information. In Fig. 2, we can observe the model benefits when syntactic information is introduced and the percentage of it does not exceed 80%. It has indicated that syntactic information is more important than semantic information in such translations and either type of information is indispensable. This discovery may also encourage more researchers to focus on how to better utilize the syntactic information of sentences to help translations. And we will set p to 60% for the rest of experiments.

Table 5. Comparison of our method and existing methods.

Method	BLEU
Previous methods	
Zhang et al. [27]	24.10
Chang et al. [5]	33.10
General translation methods	
Transformer [21]	34.88
Transparent [4]	34.85
MUSE [28]	35.23
Admin [17]	35.30
R-Drop [24]	33.53
Ours	**35.88**

Table 6. Comparison of our method and two general translation engines.

Methods	BLEU
Baidu API	23.26
Microsoft API	34.68
Ours	**35.88**

5.3 Results

Comparisons with Existing Methods. For a more complete comparison, we compare two previous works on translation from classical to modern Chinese and some SOTA methods [17,24,28] for other language pairs. Bapna et al. [4] used the output of each encoder layer like us, but in a simple summation way. From Table 5, we can see that our method has achieved the best BLEU score **35.88**, which is **1.0**↑ BLEU higher than vanilla Transformer. What's more, to our surprise, R-Drop, which achieved SOTA on several translation tasks, performs poorly in translation from classical to modern Chinese. We speculate that the reason behind this is we do not have an over-fitting problem here, which leads to the under-fitting.

Comparisons with Translation Engines. For a further and fairer comparison, we compare two general translation engines, Baidu[4] and Microsoft[5], which are currently available to provide classical to modern Chinese translations. We translated the test set by pulling their APIs and scored the resulting translations. The results are given in Table 6. Our model can achieve a **35.88** BLEU score which is **12.62**↑ and **1.20**↑ better than Baidu and Microsoft API respectively. Some translations of different methods are shown in Table 4. It can be seen that our method can translate the sentence relatively completely, while the two translation engines show an obvious lack of fluency and completeness.

6 Conclusion

In this paper, we aim to obtain better neural machine translations from classical to modern Chinese. We present the first high-quality and comprehensive dataset named C2MChn for the translation from classical to modern Chinese. We believe that the public release of this dataset will greatly benefit the study of machine translation from classical to modern Chinese. Furthermore, we propose a simple yet effective method that can effectively exploit syntactic and semantic information from classical Chinese sentences. Through this method, we have demonstrated that both syntax and semantics are indispensable in the translation from classical to modern Chinese. Empirical experiments have shown the superiority of our approach by comparing it with several SOTA methods and two general translation engines. In the future, we hope to further enrich the domains of our dataset and explore how to better fuse both syntactic and semantic information for better translations.

Acknowledgement. This research is supported in part by NSFC (Grant No.: 61936003) and Zhuhai Industry Core and Key Technology Research Project (no. 2220004002350). We would like to thank Mr. Xiandu Shi and Ms. Jing Zhang for providing some original data collation and data annotation for this work.

References

1. Ataman, D., Negri, M., Turchi, M., Federicob, M.: Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. Prague Bull. Math. Linguist. **108**, 331–342 (2017)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bahdanau, D., Cho, K.H., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
4. Bapna, A., Chen, M.X., Firat, O., Cao, Y., Wu, Y.: Training deeper neural machine translation models with transparent attention. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3028–3033 (2018)

[4] https://fanyi.baidu.com/.

[5] https://www.bing.com/translator/.

5. Chang, E., Shiue, Y.T., Yeh, H.S., Demberg, V.: Time-aware ancient Chinese text translation and inference. In: Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021, pp. 1–6 (2021)
6. Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q., Salakhutdinov, R.: Transformer-XL: attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988 (2019)
7. Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., Kaiser, L.: Universal transformers. In: International Conference on Learning Representations (2019)
8. Dou, Z.Y., Tu, Z., Wang, X., Shi, S., Zhang, T.: Exploiting deep representations for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4253–4262 (2018)
9. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning, pp. 1243–1252. PMLR (2017)
10. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 1631–1640 (2016)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Hurskainen, A., Tiedemann, J.: Rule-based machine translation from English to Finnish. In: Proceedings of the Second Conference on Machine Translation, pp. 323–329 (2017)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
14. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: International Conference on Learning Representations (2019)
15. Kontogianni, A., Ganetsos, T., Kousoulis, P., Papakitsos, E.C.: Computer-assisted translation of Egyptian-Coptic into Greek. J. Integr. Inf. Manage. (2020)
16. Liu, D., Yang, K., Qu, Q., Lv, J.: Ancient-modern Chinese translation with a new large training dataset. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP) **19**(1), 1–13 (2019)
17. Liu, L., Liu, X., Gao, J., Chen, W., Han, J.: Understanding the difficulty of training transformers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 5747–5763 (2020)
18. Raganato, A., Tiedemann, J., et al.: An analysis of encoder representations in transformer-based machine translation. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. The Association for Computational Linguistics (2018)
19. So, D., Le, Q., Liang, C.: The evolved transformer. In: International Conference on Machine Learning, pp. 5877–5886. PMLR (2019)
20. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
22. Wang, Q., et al.: Learning deep transformer models for machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1810–1822 (2019)
23. Wang, Q., Li, F., Xiao, T., Li, Y., Li, Y., Zhu, J.: Multi-layer representation fusion for neural machine translation. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3015–3026 (2018)

24. Wu, L., et al.: R-Drop: regularized dropout for neural networks. Adv. Neural. Inf. Process. Syst. **34**, 10890–10905 (2021)
25. Xiong, R., et al.: On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10524–10533. PMLR (2020)
26. Zhang, H., Yang, M., Zhao, T.: Exploring hybrid character-words representational unit in classical-to-modern Chinese machine translation. In: 2015 International Conference on Asian Language Processing (IALP), pp. 33–36. IEEE (2015)
27. Zhang, Z., Li, W., Su, Q.: Automatic translating between ancient Chinese and contemporary Chinese with limited aligned corpora. In: International Conference on Natural Language Processing and Chinese Computing, pp. 157–167 (2019)
28. Zhao, G., Sun, X., Xu, J., Zhang, Z., Luo, L.: MUSE: parallel multi-scale attention for sequence to sequence learning. arXiv preprint arXiv:1911.09483 (2019)

Imitation Attacks Can Steal More Than You Think from Machine Translation Systems

Tianxiang Hu[1], Pei Zhang[2], Baosong Yang[2](✉), Jun Xie[2], and Rui Wang[1](✉)

[1] Shanghai Jiao Tong University, Shanghai, China
{hutianxiang,wangrui12}@sjtu.edu.cn
[2] Alibaba Group, Hangzhou, China
{xiaoyi.zp,yangbaosong.ybs,qingjing.xj}@alibaba-inc.com

Abstract. Attackers can easily steal the capabilities of a machine translation (MT) system by imitation attack without too much cost. However, few works pay attention to this topic. In this paper, we explore when and why the MT model can be stolen. We first empirically analyze imitation attacks and model stealing on MT tasks, finding that imitation attacks can steal the victim model from noisy query data, noisy models, and noisy translations, which are the typical methods for model defense. What's more, the performance of the imitation model may even exceed the victim. By defining a KL distance of different corpora and using it to measure the similarity between the original data and stolen translations, we show that the imitation model steals MT systems relying on indirectly learning the distribution of the original data.

Keywords: Imitation attack · Machine translation · Model defense

1 Introduction

An attacker can easily steal the capability of a victim model by querying APIs to obtain synthetic corpora and reconstruct an imitation model without the victim structure, parameters, training data [6,11,22]. This is called imitation attack, also known as model stealing or model extraction [20]. These victim models would have cost companies lots of budget for human translation. However, the imitation attacks can avoid the cost, simply by using a large number of query monolingual data to obtain the synthetic corpus generated from the victim model for training their own models [23]. In addition to the above intellectual property issues, imitation attacks may also lead to the leakage of private data or be used by malicious people to generate adversarial examples to attack the original model, etc. [19].

T. Hu and P. Zhang—Contributed equally. Work was done when Tianxiang Hu was interning at Alibaba Group.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 400–412, 2023.
https://doi.org/10.1007/978-3-031-44693-1_32

Imitation attack is closely related to the well-studied knowledge distillation [7,14,25], which aims to train a student model to imitate the predicted distribution probabilities of a teacher under its guidance. However, imitation attacks differ from knowledge distillation because the training data and hidden representations of the victim model are usually unknown and the attacker cannot get the output probabilities of the victim as black-box APIs. Thus, the training process of imitation models can acquire little guidance or knowledge of victims such as KL divergence distribution matching losses.

Defenses against imitation attacks are mainly divided into stealing defense and watermarking defense [5,12,13,15,22]. Stealing defense attempts to directly prevent attackers from learning the victim models by outputting modified or noisy predictions without much performance reduction for users. Adding perturbation or noise to the model's output, the performance of the imitation model learned by the attacker on the perturbed output will be significantly reduced. Previous work [13,15,22] has mainly researched truncating posteriors, perturbing posteriors, prediction poisoning, etc.

However, such methods often lead to poor user experience for performance. In addition to stealing defense, the other is watermarking defense [5,12]. It mainly inserts backdoors into networks to achieve a post-hoc verification of ownership. When suspecting that some models are obtained by stealing, the victim can verify the possibility of being stolen by certain special query data. But if the attacker does not make the stolen imitation model public, but uses it privately, it cannot be detected. Therefore, the watermarking defense cannot prevent the model from being stolen.

In this work, we focus on the imitation attacks on neural machine translation tasks, which have high commercial value. To the best of our knowledge, there are few works focused on this topic [22]. When and why imitation attacks can steal successfully from the victim model has not been extensively explored. We study these questions more closely, conducting four sets of empirical study to answer the following questions:

- How imitation attack is affected by the domain quality or the data size of the query data?
- Why the performance of the model obtained by imitation attack stealing can exceed the victim model?
- What is the efficacy of adding noise to the decoding or modifying victim model as a defense to prevent model imitation stealing?
- Why imitation attack can steal models from noisy low-quality translations?

We show that the imitation model can steal MT systems from noisy query data, noisy models and noisy translations relying on indirectly learning the word alignment probability distribution of high-frequency words on the whole corpus.

2 Experimental Setup

2.1 Dataset

We trained and evaluated models on standard machine translation datasets which have been widely used in previous papers. We conducted experiments on the IWSLT14 German (De)-English (En), IWSLT14 Spanish (E s)-English (En) [3], WMT14 De-En and WMT14 France (Fr)-En [1]. The detailed data statistics and descriptions are shown in Table 1. For WMT14, we randomly selected 4M parallel sentence pairs for train respectively. For IWSLT14, the development sets are randomly sampled from IWSLT14 train sets and we used IWSLT14.ted.dev2010, IWSLT14.tedx.dev2012, IWSLT14.ted.tst2010, IWSLT14.ted.tst2011, IWSLT14.- ted.tst2012 as test sets. For WMT14 De-En and WMT14 Fr-En, we used newstest2013 as the development set and newstest2014 as the test set.

Table 1. The detailed data statistics and descriptions about the number of parallel sentences in each dataset.

Corpus	Language	Domain	Data size		
			train	dev	test
WMT14	De ↔ En	News	4M	3K	3K
WMT14	Fr ↔ En	News	4M	3K	3K
IWSLT14	De ↔ En	TED	160K	7K	7K
IWSLT14	En ↔ Es	TED	180K	8K	6K

2.2 Implementation Details

For the imitation attack, We first queried the victim model with the source monolingual sentence. We regard the translations as the labels of monolingual query sentences. Last we trained an imitation model on the synthetic bilingual corpus. In our following experiments, unless otherwise specified, the victim model refers to the model trained on the IWSLT14 De-En dataset and the query data is in-domain data.

For pre-processing, we used Moses tokenizer [10] including tokenization, normalize-punctuation, and cleaning. For imitation attack experiments on IWSLT14, the byte-pair encoding [18] for IWSLT14 De-En and IWSLT14 En-Es are respectively trained with 10k/10k merge operations and source-target vocabularies are shared. We used IWSLT14 De-En bpecode to process out of domain (OOD) query data WMT14 De. For imitation attack experiments on WMT14, the byte-pair encoding for WMT14 De-En and WMT14 Fr-En are respectively trained with 32k/32k merge operations and source-target vocabularies are shared. The stolen data will re-learn bpecode for training the imitation model.

2.3 Training and Hyperparameters

We implemented a series of experiments with fairseq toolkit [16] and followed the same hyper-parameter setting of base Transformer [21] for the victim model and the imitation model. All the experiments are applied with the share-embedding setting and trained on 2 GTX 1080Ti GPUs with batch size of 4096 tokens. We used Adam [9] optimizer with an initial learning rate of 0.0005, $\beta_1 = 0.9, \beta_2 = 0.98$ and the inverse_sqrt learning rate schedule. We performed early stopping if the BLEU score on the development set does not improve in 15 consequent checkpoints. We used beam search with a beam size of 4 and length penalty of 0.6 and report case-insensitive BLEU [17] on IWSLT14 test sets and case-sensitive BLEU on WMT14 test sets.

3 Effect of Query Data

3.1 Settings

In our first set of experiments, we study the effect of query data type and size. Following the settings [22], we used in-domain data (IWSLT14 De-En) to train a victim and then obtain synthetic bilingual corpora by querying the victim with monolingual German sentences. We add noise by dropout, replacing with UNK token and shuffling 20% of input tokens.[1] Random generation is to use the OOD (WMT14 De-En, randomly selected 1M parallel sentence pairs) dictionary, then select words with equal probability to generate a sentence of 15∼25 length. Random generation is an extreme case we consider, using meaningless data as a query, observing whether the victim can be stolen.

3.2 Results and Analysis

The effect of the query data type is presented in Table 2. We can see that regardless of using out-of-domain or in-domain as query data, the attacker can steal the victim model. When using query data with noise, the victim can also be stolen to a certain extent, but the effect will decrease. In the most extreme case, when using completely randomly generated sentences as query data, the imitation model can still reach 19.78 BLEU on the in-domain testset and 15.44 BLEU on the out-of-domain testset. Table 3 shows that as the data size increases, the performance of the imitation model will be improved. Furthermore, when using out-of-domain query data, more data may be needed to better steal the model.

4 Why Can Surpass Victim Model

4.1 Settings

The previous experiments show that when the query data is consistent with the training data of the victim model, the imitation model may outperform the

[1] Adding noise is performed using scripts from https://github.com/jxhe/self-training-text-generation.

Table 2. The performance (BLEU) of imitation model on different types of query data under In-domain and OOD testset. Imitation model steals from different types of query data.

Query Data Type	In-domain	OOD
Victim Model	34.79	19.96
In-domain	35.02	20.38
OOD	33.06	19.90
OOD + Noise	32.48	20.08
OOD + Random	19.78	15.44

Table 3. The performance (BLEU) of imitation model with different query data size under in-domain testset.

Query Data Size	40k	80k	160k
In-domain	31.12	33.26	35.02
Query Data Size	160k	500k	1000k
OOD	29.12	32.21	33.06
Random	15.15	18.53	19.78

victim. The public corpus always has noise [2], so we hypothesize that stealing this distillation-like method can reduce the data noise in the victim and make the corpus smoother.

We use the same experimental settings in Sect. 3. First, we verify that the improvement of the stealing model is not sustainable through a simple experiment. As shown in Table 4, we iteratively stole the imitation model obtained by stealing, and find that the performance is improved only at the first stealing. Then we performed the stealing experiments on the victim models trained with adding noisy corpus. The method of adding noise is the same as in the previous section. The stealing uses the in-domain data as query data.

4.2 Results and Analysis

As presented in Table 5, we can see that when the training data contains extra noise, the performance of the victim will drop and the imitation model obtains more improvement. We think the imitation attack can make the noisy data smoother and recover some information. Then imitation model will have a better performance than the victim under more high-quality training data.

5 Efficacy of Model Defense

In this section, we explore the effect of some steal defense strategies for anti-stealing. To the best of our knowledge, we found that there are not many defense

Table 4. Results of iterative stealing. Steal*2 denotes we steal the Steal model, and Steal*3 denotes we steal the Steal*2 model. Performance (BLEU) only improves at first stealing.

Model	Victim	Steal	Steal*2	Steal*3
In-domain	34.79	35.02	34.97	34.95
OOD	19.96	20.38	20.46	20.37

Table 5. Experimental results (BLEU) on noisy victim models. We add noise on half of the data at the source or target when training victim. Imitation model obviously surpasses victim when more noise in training data.

Data	Model	In-domain	OOD	Δ
Original	Victim	34.79	19.96	-
	Steal	35.02	20.38	+0.33
Tgt + Noise	Victim	32.76	17.88	-
	Steal	33.70	19.37	+1.22
Src + Noise	Victim	33.44	18.53	-
	Steal	34.40	20.12	+1.28

strategies for machine translation. Some of the strategies used in previous classification tasks are not applicable to autoregressive tasks. Therefore, we adopted some intuitively feasible methods based on previous defense strategies used in classification tasks to perturb the output.

1) Noisy Decoding Strategy (NDS): The idea of this defense is the victim does not output the highest-quality translation, but sentences such as sampling top 20 [8] whose translation probability score is not very high. We hope that the imitation attack cannot learn the original model well on the noisy translations. Beam Top1: set the beam size to 4, and select the translation with the highest translation score. Sampling: set the beam size to 20, and use the sampling decoding, then select Top1, Top10, and Top20 according to the translation probability. The victim model generally uses beam search to output the translation, but we use sampling decoding as a method of adding noise.

2) Round Trip Translation (RTT): The idea is to use a third-party language, such as Spanish (Es) in our experiment, to translate the translation into Spanish through the En-Es translation model, and then translate it back to English through the Es-En translation model [4]. For RTT in detail, we first trained the translation models of En-Es and Es-En with the ISELT14 En-Es corpus. For each German query data, we performed beam search decoding. Then we translated the output text into Spanish through the En-Es translation model and translated it back to English through the Es-En model. Through the Round Trip Translation, we hope to protect the original translation output.

3) Model Modification (MM): The idea of this method is to modify the original model. Early Stop Victim refers to selecting an epoch in the training of

Table 6. Randomly sampled IWSLT14 pair with synthetic translations. The translation quality of Sampling Top 20 and Early Stop Victim are quite poor.

Source	sie laufen aber auch der gefahr, einen vitamin-d-mangel zu entwickeln, wenn sie büroarbeit machen, so wie dieser typ
Reference	they also run the risk of vitamin d deficiency, if they have desk jobs, like that guy
Beam Search	but they also run the risk of vitamin d deficiency when they're doing office work, like this guy
Sampling Top20	but they also run überraschexacgel into the danger of vitamin d in the beydevice once gesamiting the office product – like this guy
Early Stop Victim	but they're going to go to the danger, a vivivivic, if they do work likethis guy

the victim model. Slightly Damaged and Heavy Damaged indicate victim model parameters are damaged to varying degrees (randomize some parameters). For MM in detail, the idea here is not to protect the model. We want to explore whether the imitation attack can steal information from the damaged model. Slightly Damaged is to randomize the parameter of encoder_attn.out_proj.weight in the first decoder layer between $-0.5\sim0.5$. Heavily Damaged is to randomize the parameter of encoder_attn.out_proj.weight in the first and last decoder layer between 0.5–0.5.

5.1 Results and Analysis

We conduct experiments on IWSLT14 De-EN, WMT14 De-En and WMT14 Fr-En three datasets. For WMT14 De-En and WMT14 Fr-En, we randomly selected 2M parallel sentence pairs for train dataset as query dataset, the remaining 2M data is used as the training set of the victim model.

IWSLT14 De-En. The effect of simple defense is presented in Table 7. By observing the two columns of **Victim** and **Steal Model**, we can find that almost all imitation stealing models perform better than victim models with defenses. This does not reach our expectation for the effect of model defense against imitation attack, the simple model defense strategies do not work. **Ref-Train-Similarity** indicates the similarity between the victim train data (reference) and the stolen translations. It can be found that the translations generated by the sampling top20 are low quality (only 17.39 BLEU compared to sampling top1, Table 6 shows a example of low quality synthetic translation). But according to the column of **Steal Model**, we can see that even if the imitation model is trained on poor translations like the sampling top20 (11.10 BLEU), the stealing model can still reach 31.03 BLEU. This means the imitation model can better learn the victim model from data generated from the defense model. However, not all poor translations obtained by the victim model can train a good imitation

Table 7. The effect of different kinds of simple defense strategies against imitation attacks. Measure the similarity between the victim train data (reference) and stolen translations as Ref-Train-Similarity. Evaluate the defense victim and steal model with BLEU. Imitation attacks can steal from noisy translations.

Method	Defense Method	Ref-Train-Sim		Defense Victim		Steal Model	
		BLEU	BERTScore	In-domain	OOD	In-domain	OOD
NDS	Beam Top1	54.13	0.75	34.79	19.96	35.02	20.38
	Sampling Top1	44.82	0.69	29.10	15.83	33.88	19.55
	Sampling Top10	31.34	0.49	20.37	10.85	32.95	18.77
	Sampling Top20	17.39	0.18	11.10	6.65	31.03	17.56
RTT	De→En→Es→En	28.65	0.55	29.20	15.25	32.31	18.09
MM	Early Stop Victim	16.98	0.39	16.10	5.63	16.51	5.61
	Slightly Damaged	51.71	0.72	33.05	18.78	34.54	20.44
	Heavily Damaged	7.73	0.19	5.80	1.81	2.94	0.78

Table 8. The effect of different kinds of simple defense strategies against imitation attacks on WMT14 De-En dataset. Imitation attacks can learn and steal from a poor translations.

Method	Defense Method	Defense Victim		Steal Model	
		BLEU	BERTScore	BLEU	BERTScore
NDS	Beam Top1	30.09	0.68	29.51	0.68
	Sampling Top1	20.52	0.45	28.02	0.67
	Sampling Top10	12.28	0.23	26.97	0.65
	Sampling Top20	3.19	-0.08	18.33	0.55
MM	Early Stop Victim	19.40	0.51	20.11	0.64

Table 9. The effect of different kinds of simple defense strategies against imitation attacks on WMT14 Fr-En dataset. Imitation attacks can learn and steal from a poor translations.

Method	Defense Method	Defense Victim		Steal Model	
		BLEU	BERTScore	BLEU	BERTScore
NDS	Beam Top1	34.62	0.71	33.82	0.71
	Sampling Top1	23.26	0.47	32.87	0.69
	Sampling Top10	13.95	0.27	31.65	0.68
	Sampling Top20	3.62	0.01	23.29	0.60
MM	Early Stop Victim	27.21	0.61	26.90	0.65

model. The results of Early Stop Victim show when the Ref-Train-Similarity is 16.89 BLEU, the stealing model only gets 16.51 BLEU.

WMT14 De-En and WMT14 Fr-En. We can see from Table 8 and Table 9 that although simple defense strategies do not prevent model stealing effectively, a interesting phenomenon is the stolen model can be successful in stealing from a corpus that looks poor at the sentence level (as shown in Table 8, the translation quality of the sampling top20 is only 3.19 BLEU, and the stealing model has learned a translation model of 18.33 BLEU). But on the other hand, under the defense strategy of early stop, the above phenomenon does not appear in the stealing model.

6 Why Can Steal from Noisy Data

In this section of experiments, we show the imitation model steals the victim from learning the distribution of the original data. We find that on some poor translations (sampling top20), a good imitation model can be trained, but some cannot (Early Stop Victim). We think that BLEU or BERTScore [24] can't explain the quality of translations very well.

6.1 KL Score

We think the main performance of the machine translation model depends on the translation of high-frequency words. Although the translation of the sampling top20 is poor, its word alignment probability distribution of high-frequency words on the whole corpus is correct. We define a KL score to evaluate the quality of stolen bilingual translations. We aligned the source and target words of the corpus through word alignment tools[2], selected 5K words with the highest frequency in the German corpus, and calculated an alignment probability vector for each word. Finally we calculated the average word-aligned KL score of the 5K words between the stolen translation corpus and the original victim corpus as follows:

$$\mathrm{KL}(v,t) = \sum_{x \in \mathcal{V}_x} \sum_{y \in \mathcal{V}_y} p_v(y|x) \ln \frac{p_v(y|x)}{p_t(y|x)}, \tag{1}$$

where p_v denotes the alignment probability estimated using the victim training data and p_t denotes the alignment probability estimated using the stolen bilingual translations, x and y denote a word in the source and target vocabulary respectively, $\mathcal{V}_x$ is a set of the 5K words with the highest frequency in source corpus, and $\mathcal{V}_y$ is target vocabulary.

There is an example for computing the KL similarity between two corpora. Consider the German word “sehr”, which can correspond to multiple words in English, such as “very”, “really”, and “quite”. After aligning the original corpus using a fast-align or eflomal alignment model, we can obtain the probability of each target word corresponding to “sehr”, such as p(“very”—“sehr”) = 0.730, p(“really”—“sehr”) = 0.063, and p(“quite”—“sehr”) = 0.014. Then, we select the 5,000 most frequent words from the German corpus as candidates for x, and use the entire target vocabulary Y as candidates for y. Each x corresponds to a vector, such as p(Y—“sehr”) = [0.730, 0.063, 0.014, ...]. For the stolen bilingual

[2] https://github.com/robertostling/eflomal/.

corpus, we can obtain an alignment vector for “sehr” in the same way. If a word y in the vocabulary Y does not appear in the stolen corpus, such as “quite” not appearing in the stolen corpus as a translation of “sehr”, we set its alignment probability to a small value such as 1e−5.

6.2 Results and Analysis

As shown in Table 10, even though the translation quality of sampling top20 is poor, the KL distance between it and the victim training corpus is small. The translation of a single sentence may poor, but from the perspective of the entire corpus, the translations contain the alignment information of original training data. On the contrary, the KL distance of Early Stop Victim is relatively large, so it is hard to learn a better imitation model on this translation. The Early Stop Victim model has not learned well in terms of word alignment. Translations generated by this model contain little information about the original data.

Table 11 shows why out-of-domain data can also be used to steal the victim model from the perspective of KL. The translations obtained by the victim model will be closer to the training corpus of the victim (0.94 → 0.67), and farther away from its own domain (0.67 → 1.31). We think whatever the query data domain is,

Table 10. The relationship of imitation model’s performance (BLEU) and KL distance on in-domain query data. Ref-Train-KL denotes the KL distance of the original IWSLT14 corpus and the stolen corpus.

Method		Steal-Test	Ref-Train-KL
IWSLT14 De-En	Beam Top1	35.02	0.39
	Sampling Top20	31.03	0.53
	Early Stop Victim	16.51	1.00
WMT14 De-En	Beam Top1	29.51	0.05
	Sampling Top20	18.33	0.03
	Early Stop Victim	20.11	0.18
WMT14 Fr-En	Beam Top1	33.82	0.04
	Sampling Top20	23.29	0.05
	Early Stop Victim	26.90	0.19

Table 11. The KL similarity between different corpus. WMT14 beam denotes the synthetic bilingual corpus by querying monolingual WMT14 sentences through victim model. The translations will be closer to the victim training data and farther away from its own domain.

KL Score	IWSLT14	WMT14
WMT14	0.94	0.00
WMT14 beam	0.67	1.31

the output translations will leak the word alignment information of the original training data. Therefore, the capability of the original victim model is stolen by stealing word alignment distribution information.

7 Conclusion

We study imitation attacks against MT tasks and find that imitation attacks have a strong ability to steal model knowledge from noisy query data or noisy models. When we adopt simple noise defense strategies, attackers can still steal from these noisy translations. We verify why the imitation attack can steal the victim from several angles: (1) The imitation attack can make the stolen data smoother than the original training data; (2) The imitation attack relies on indirectly learning the word alignment distribution of original data. The translations generated by the victim will be closer to the domain of victim training data, not itself domain.

Acknowledgements. This project is mainly supported by the Alibaba-AIR Program (22088682). Tianxiang and Rui are with MT-Lab, Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200204, China. Rui is also supported by the General Program of National Natural Science Foundation of China (6217020129), Shanghai Pujiang Program (21PJ1406800), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and Beijing Academy of Artificial Intelligence (BAAI) (No. 4).

References

1. Bojar, O., et al.: Findings of the 2014 workshop on statistical machine translation. In: Proceedings of the Ninth Workshop on Statistical Machine Translation (2014)
2. Briakou, E., Carpuat, M.: Can synthetic translations improve bitext quality? In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2022)
3. Cettolo, M., et al.: Report on the 11th IWSLT evaluation campaign. In: Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign (2014)
4. Haffari, G., Roy, M., Sarkar, A.: Active learning for statistical phrase-based machine translation. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2009)
5. He, X., et al.: Cater: Intellectual property protection on text generation apis via conditional watermarks. In: Advances in Neural Information Processing Systems (2022)
6. He, X., Lyu, L., Sun, L., Xu, Q.: Model extraction and adversarial transferability, your BERT is vulnerable! In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2021)

7. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. ArXiv (2015)
8. Holtzman, A., et al.: The curious case of neural text degeneration. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020 (2020)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
10. Koehn, P., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions (2007)
11. Krishna, K., et al.: Thieves on sesame street! model extraction of bert-based apis. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020 (2020)
12. Li, Y., et al.: Untargeted backdoor watermark: towards harmless and stealthy dataset copyright protection. In: Advances in Neural Information Processing Systems (2022)
13. Mazeika, M., Li, B., Forsyth, D.A.: How to steer your adversary: targeted and efficient model stealing defenses with gradient redirection. In: Proceedings of the 39th International Conference on Machine Learning (2022)
14. Mobahi, H., Farajtabar, M., Bartlett, P.L.: Self-distillation amplifies regularization in hilbert space. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 6–12 December, 2020, virtual (2020)
15. Orekondy, T., Schiele, B., Fritz, M.: Prediction poisoning: Towards defenses against DNN model stealing attacks. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April, 2020 (2020)
16. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (2019)
17. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (2002)
18. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2016)
19. Szegedy, C., et al.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April, 2014, Conference Track Proceedings (2014)
20. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction apis. In: USENIX Security Symposium (2016)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December, 2017, Long Beach, CA, USA (2017)
22. Wallace, E., Stern, M., Song, D.: Imitation attacks and defenses for black-box machine translation systems. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
23. Xu, Q., et al.: Student surpasses teacher: Imitation attack for black-box NLP APIs. In: Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju, Republic of Korea (2022)

24. Zhang, T., et al.: Bertscore: evaluating text generation with BERT. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April, 2020 (2020)
25. Zhou, C., Gu, J., Neubig, G.: Understanding knowledge distillation in non-autoregressive machine translation. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April, 2020 (2020)

Monolingual Denoising with Large Language Models for Low-Resource Machine Translation

Haoyu Xu[1], Xing Wang[2], Xiaolin Xing[1], and Yu Hong[1](✉)

[1] Soochow University, Computer Science and Technology, Suzhou, China
tianxianer@gmail.com
[2] Tencent AI Lab, Shenzhen, China

Abstract. Low-resource machine translation struggles over the issue of bilingual data sparsity. Self-training based bilingual data augmentation is potentially useful for overcoming the issue. However, the resultant pseudo-parallel data comprises a variety of noises in the target language, including grammatical errors, abnormal word sequences, misspellings, mistranslations, etc. The noises unavoidably cause distraction during training. In this paper, we propose to refine the pseudo-parallel data using monolingual denoising. Specifically, we finetune mBART model to low-resource parallel data and identify noisy samples by self-inspection during the self-training process. On this basis, we leverage large language models., e.g., ChatGPT, to fix the possible errors that occurred in the target language of noisy samples using manually-edited prompts. This allows the refined pseudo-parallel data to be produced. We employ the aforementioned data to augment and retrain the mBART model. We conduct experiments on benchmark low-resource English-oriented translation corpora in OPUS-100 which possess different source languages, including Georgian (Ka), Urdu (Ur), and Slovenian (Sl). Experimental results show that our method achieves substantial improvements, allowing the translation performance to reach the chrF++ scores of 36.8%, 43.5%, and 47.5%.

Keywords: Machine Translation · Low Resource · Large Langauge Model

1 Introduction

Neural Machine Translation (NMT) [2,27,33] has achieved considerable progress in recent years. However, its performance remains underperformance for low-resource language pairs compared to their high-resource counterparts, due to the lack of large-scale parallel corpora [22,30]. In the practical production scenario, low-resource neural machine translation (LNMT) also suffers from the issue of noisy training data. Consequently, the LNMT models struggle to capture and address complex linguistic phenomena, including grammatical structures, long-distance dependencies, and ambiguity [3,4].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 413–425, 2023.
https://doi.org/10.1007/978-3-031-44693-1_33

Several efforts have been made to tackle the previously mentioned challenge. These methodologies can be broadly divided into two primary categories: transfer learning and data augmentation. Transfer learning transfers high-resource NMT models to languages with limited parallel resources [37,40], or distilling multilingual translation knowledge for transfer [1,26,32]. Data augmentation is implemented by various solutions, such as paraphrasing, back translation [24], self training [10,39] and synonym replacement. These methods can produce pseudo-parallel data at a comparable scale, and are therefore practical for alleviating data sparsity, to some extent [6,7,15,36].

However, our findings reveal the fact that there are a large number of noises occurred in the pseudo-parallel data. In a pilot study of self-training based augmentation, we observe that various noisy parallel data is caused by grammatical errors, abnormal word sequences, misspellings and mistranslation, respectively. Refining the pseudo-parallel data by denoising contributes to the retraining and optimization of LNMT, though manual denoising is time-consuming.

In this paper, we propose a denoising approach in the framework of self-training based data augmentation towards LNMT. Large language models (LLMs) are utilized in our study, which have shown a remarkable ability across a wide range of natural language processing (NLP) tasks, e.g., machine translation [8,29], grammatical error correction [35] and context understanding [5,13].

Specifically, we finetune mBART to the low-resource languages for NMT. During self-training, we carry out self-inspection to evaluate mBART over the original parallel data, so as to identify the samples resulting in a lower BLEU score. We regard such samples as potential noises and perform denoising over them. Our denoising method is not to filter noises. Instead, we leverage LLMs to rewrite the target language of noisy parallel samples in the manner of auto-regressive generation, in terms of manually-edited prompts. On the basis, we conduct data augmentation using the rewritten samples and retrain mBART over the augmented data.

We experiment on the benchmark OPUS-100 [38], concentrating on the LNMT tasks of translating Ka, Ur and Sl into English (En). The experimental results show that our method substantially improves the LNMT performance, achieving the chrF++ scores of 36.8%, 43.5%, 47.5%, as well as BLEU scores of 19.6%, 23.9%, and 28.5%. In addition, we verify the effects of different techniques for denoising, including grammar error correction model T5. It is proven that the large language model ChatGPT contributes more to the denoising and subsequent data augmentation processes.

Because we want to repair the target language, the target languages we choose should have not bad performance on LLMs. Consider experiments effective, We conduct LNMT on the target language is English setting. Interestingly, the data obtained by our method in the direction translate to English also brought a great improvement when training from the direction translate from English NMT.

The rest of the paper is organized as follows. Section 2 overviews the related work. Section 3 presents the methodological framework and computational details. In Sect. 4, we introduce our experimental settings and analyze the results. We conclude the paper in Sect. 5. All codes have been released on Github[1].

2 Related Work

There are many advanced methods of LNMT, but due to length limitations, we will only briefly review the relevant methods in recent years.

Data Augmentation. Data augmentation has been shown to be effective in alleviating the data scarcity problem. The central idea of data augmentation is artificially increasing the training set by creating modified copies of a dataset using existing data. In machine translation research, techniques such as self-training [10,39] and back-translation [24] leverage monolingual data from either the source or target languages to synthesize pseudo-bilingual data. SwitchOut [34] is a method that entails the random replacement of selected words within both the source and target sentences using alternative terms from their respective vocabularies. Data diversification [16] seeks to enrich training data by applying forward-translation to the source language within the parallel data or implementing back-translation to the target language in a reverse translation direction. Data rejuvenation [9] aims to identify and reutilize the inactive training samples to enhance translation performance.

Data Denoising. The significance of data denoising in NMT has garnered increasing attention. To this end, MT community organized a parallel corpus filtering task in the Conference on Machine Translation (WMT) 2018 [12]. In machine translation research, a variety of approaches have been explored. The paper [11] explored the effects of five types of artificial noise present in parallel data on NMT training, concluding that NMT displays relatively low robustness to data noise. This paper [28] carried out a comprehensive empirical investigation of semantic divergences and proposed to select high-quality translation examples by identifying semantic divergences in parallel data. The work [31] proposed to measure noise in parallel data and select noise-reduced data to train NMT.

3 Approach

In this section, we first describe the term definition employed in the subsequent sections. Then we provide a detailed description of the overall process. Finally, we present the proposed approach that leverages LLMs for monolingual denoising tasks in low-resource neural machine translation.

3.1 Preliminary Study

To investigate the issue of noise in self-training, we conducted a series of comparative experiments in high-resource and low-resource settings. Specifically, we

[1] https://github.com/XDeepAzure/nmt-corrector-src.

empirically selected French (Fr)→English (En) as the high-resource setting and Georgian (Ka) →English (En) as the low-resource setting. To ensure a fair comparison, we used the state-of-the-art multilingual translation model nllb-200-distilled-1.3B [17] model. Subsequently, we employed model to translate the source side of the parallel training data. We utilized BLEURT [23] to assess the similarity between the translated outputs and their corresponding target sentences. The detailed experimental settings are introduced in Sect. 4.

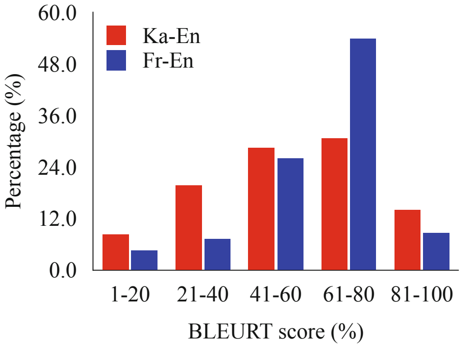


Fig. 1. Ratio of examples that are grouped by BLEURT score. Sample 3k sentences from the Fr-En, Ka-En train set in OPUS-100 [38].

The pair examples, consisting of the translation output and the target sentence, were divided into five bins based on their BLEURT scores. The percentage distribution of each bin is presented in Fig. 1. We find that low-resource Ka→En has more low BLEURT score examples than Fr→En, demonstrating that the self-training in the low-resource machine translation has a severe noise problem. This motivates us to perform monolingual denoising with large language models for low-resource machine translation.

3.2 Term Definition

In this section, we describe the terms used in the following sections.

- ***base model*:** The LNMT base model is trained by available parallel corpus without applying the proposed method.
- ***src, tgt*:** refer to source sentence (*src*), target sentence (*tgt*) corresponding to source sentence in the parallel corpus.
- ***pred*:** refer to the predicted target generation which is generated by the trained *base model*.
- ***cor*:** refer to the revised text which is obtained by LLMs to revise the ill-generated *pred*.
- ***final model*:** refer to use the refined parallel data to train the final model.

3.3 Overall Process

The process of the proposed approach is shown in Fig. 2, which is composed of four steps:

- ***Model training.*** We use the parallel data (*src, tgt*) to train the LNMT model by using bilingual data to fine-tune the mBART-Large-50 [26] model;
- ***Target Generation.*** We utilize the trained LNMT model to perform the self-training on the source sentences and obtain the target generation outputs *pred*. Next we identify the ill-generated target output from *pred*;
- ***Target Denoising.*** We employ the LLMs to rectify the ill-generated target outputs identified in the *Target Generation* process and obtain the revised sentences *cor*;
- ***Target Reuse.*** We use the refined parallel data (*src, cor*) to train the final LNMT model.

The key idea of the proposed approach is that we first use the self-training to obtain the low-quality initial translation output and then utilize the large language model to revise the translation output.

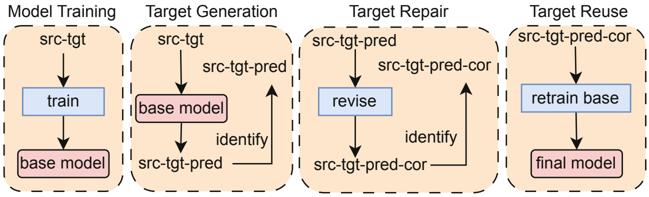


Fig. 2. Process of the proposed approach. “src”, “tgt”, “pred”, “cor” denote respectively source sentence, target sentence, target prediction output, and revised target prediction output.

3.4 Monolingual Denoising with Large Language Models

Identification. Motivated by the previous study on corpus filtering [14], we propose a simple heuristic to implement the identification process of the ill-generated target translation outputs. Given the source sentence *src*, we consider the generated target sentence as ill-generated:

$$\alpha < Sim(pred, tgt) < \beta \tag{1}$$

where the $Sim()$ is the similarity function that measures the similarity between the translation output *pred* and the corresponding target sentence *tgt*, and both α and β are pre-defined thresholds.

Table 1. Example of a denoising output with ChatGPT.

Prompt	*Correct common sense errors, grammatical structure errors, and word order errors in the following sentences, and return the corrected sentence:*
Input	*It was discovered upo_n hearing phone call the that his family is in Kuwait.*
Output	*It was discovered upon hearing a phone call that his family is in Kuwait.*

Denoising. After collecting the ill-generated target outputs, we follow the paper [8] to design prompts to rectify the ill-generated target outputs. The prompt is shown in Table 1. As shown in the table, the ill-generated translation output is revised by a word spelling ("upo_n" to "upon"), insertion of an indefinite article ("hearing phone" to "a hearing phone"), and a word deletion ("call the that" to "call that").

4 Experiment

In this section, we conduct experiments on low-resource translation tasks to verify the proposed approach. We first introduce datasets and evaluation metrics. Next, we describe the baseline and comparison systems and the hyperparameter settings. Then we list the experimental results. Finally, we conduct ablation experiments to evaluate the effectiveness of each component from the proposed approach.

4.1 Datasets and Evaluation Metrics

Datasets. We conduct machine translation experiments on the OPUS-100 corpus [38][2], wherein we focused on low-resource machine translation tasks involving three distinct language families: Ka→En, Ur→En, and Sl→En. Specifically, the designation Ka signifies the Georgian language belonging to the Kartvelian language family, Ur represents Urdu belonging to the Indo-European language family, and Sl denotes Slovenian as part of the Balto-Slavic language family. Table 2 presents the Statistics of the datasets, along with the corresponding language family of the source language.

Table 2. Statistics of the datasets used for training, valid and test sets in OPUS-100.

Language Family	Language Pair	Train	Dev	Test
Kartvelian	En-Ka	0.37M	2K	2K
Indo-European	En-Ur	0.74M	2K	2K
Balto-Slavic	En-Sl	1.00M	2K	2K

[2] https://opus.nlpl.eu/opus-100.php.

Evaluation Metrics. We adopt the BLEU, chrF++, TER as the evaluation metric for translation performance. Besides, we employ a representation-based metric, BLEURT, to evaluate the performance.

- **BLEU** [18] is an n-gram based automatic evaluation metrics. We report detokenized case-sensitive SacreBLEU [21][3], with default tokenizer settings.
- **chrF++** [20], which is an extension of chrF [19], considers the recall rate of 1-gram and 2-gram words.
- **TER** [25], also known as translation error rate, evaluates the edit distance between the translated sentence and the reference translation.
- **BLEURT**[4] [23], which employs a neural network model to score translation outputs. It can model human assessment with superior accuracy.

To compute the BLEU, chrF++, and TER scores, we utilize the SacreBLEU toolkit, following established practices in previous research [26].

4.2 Systems

To obtain a robust baseline system, we follow the common practice to finetune mBART-Large-50 [26] model on the OPUS-100 bilingual corpus. Besides, we adopt various state-of-the-art translation systems for comparison.

- **ChatGPT**[5]. Following the work [8], we directly use ChatGPT to perform the translation task with the prompt provided by the work [8].
- **mBART-M-to-M** [26]. mBART-M-to-M is a multilingual model finetuned the mBART-Large-50 on English-centric parallel corpora with 50 translation directions.

As for the proposed approach, we use the finetuned BART-Large-50 as the pre-trained model to translate the source sentences in the training data. We set the hyper-parameter α to 0.4 and β to 0.8 to identify the ill-generated translation outputs. For the ill-generated outputs, we use the large language model ChatGPT with the prompt described in Sect. 3.3 to revise the output.

4.3 Hyper-parameter Settings

We use the tokenizer of PLM (pre-trained mBART-Large-50) to tokenize the data. We set the batch size to 16 for the model fine-tuning and the gradient accumulation step to 2. We set the maximum sentence length of both the source and target sentences to 256. The learning rate (lr) is 2e−5, with a weight decay rate 0.01. The early stop patience is 15.

We use the HuggingFace Transformers library[6] and AdamW for all training procedures. All training is performed on A100 40G GPU.

[3] https://github.com/mjpost/sacrebleu.

[4] https://github.com/lucadiliello/bleurt-pytorch. We use the recommended checkpoint `lucadiliello/BLEURT-20` as described in the paper.

[5] https://chat.openai.com.

[6] https://huggingface.co/.

4.4 Main Result

Table 3 lists the experimental results on the test sets. We can observe that 1) fine-tuning mBART-Large-50 on the downstream language pairs achieves strong performance compared to ChatGPT and mBART-M-to-M. 2) the proposed monolingual denoising approach can improve the baseline system mBART-Large-50 in most evaluation metrics. These results demonstrate the effectiveness of the proposed approach.

We present a translation example in Table 4. Illustrating the substantial enhancements achieved in the sentence structure and fluency of the translation.

Table 3. The translation performances on test sets. mBART-M-to-M represents to mBart-Large-50-Many-to-Many. The bold denotes the best performance.

Model	Ka→En		Ur→**En**		Sl→En	
	BLEU	chrF++	BLEU	chrF++	BLEU	chrF++
ChatGPT	5.95	11.14	15.23	38.59	26.19	47.07
MBart-M-to-M	0.60	9.36	12.66	32.40	19.93	39.53
Baseline	**19.72**	36.53	23.40	42.78	28.13	47.20
+Denosing	19.64	**36.80**	**23.88**	**43.48**	**28.52**	**47.53**
Model	Ka→En		Ur→En		Sl→En	
	BLEURT	TER↓	BLEURT	TER↓	BLEURT	TER↓
ChatGPT	30.69	114.86	52.28	79.42	64.66	66.99
MBart-M-to-M	25.08	132.21	**54.39**	80.41	56.95	76.68
Baseline	57.95	**74.09**	50.56	71.06	64.07	64.13
+Denosing	**58.37**	74.26	53.99	**67.68**	**64.13**	**63.73**

Table 4. Example of the output which is generated by the proposed monolingual denoising approach with large language models.

Baseline	*All that is in the heavens and the earth glorifieth Allah, and He is the Mighty, the Wise.*
Reference	*All that is in the heavens and the earth glorify God. He is Majestic and All-wise.*
Denoising	*All that is in the heavens and the earth glorifies Allah. He is the Almighty, the Wise.*

4.5 Ablation Study

In this section, we conduct ablation experiments with fine-tuned mBART-Large-50 model on Ur→En test set.

Identification Threshold. The key step of the proposed approach is the noise identification process described in Sect. 3.3. The noise identification, which determines whether the output is ill-generated, is a classic classification problem. The hyper-parameter α is used to balance between recall and precision. To investigate the impact of the identification threshold, we set the hyper-parameter α

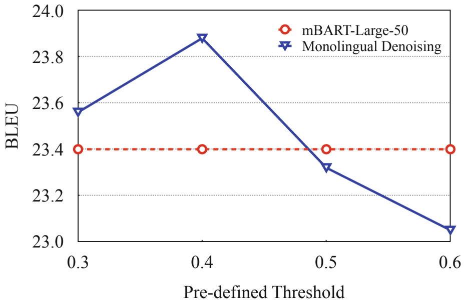


Fig. 3. Translation performance with different threshold values.

Table 5. The effect of different error correction tools. Bold is the best performance."*+w/o Denosing*" represents without denoising step.

Metric	BLEU	chrF++	BLEURT	TER↓
mBART-Large-50	23.40	42.78	50.56	71.06
+w/o Denosing	23.52	43.12	52.73	68.52
+T5-Base	21.25	39.94	51.75	70.85
+T5-Large	22.85	42.10	52.53	69.86
+ChatGPT	**23.88**	**43.48**	**53.99**	**67.68**

in Eq. (1) to {0.3, 0.4, 0.5, 0.6} to identify the ill-generated translation outputs. Then we utilize ChatGPT with the prompt described in Sect. 3.3 to rectify the ill-generated output. The experimental results are presented in Fig. 3. We can find that the proposed approach achieves the best performance with the identification threshold of 0.4.

Noise Type. Some researchers may doubt that the improvements are mainly comes from the grammatical error correction, as previous research shows that grammatical error has a great effect on translation performance. To dispel the doubt, we implement the monolingual denoising process with a grammatical error correction tool. For the grammatical error correction, we adopt T5-Large[7] is better than T5-Base[8] as the correction tool to revise the ill-generated translation output. Experimental results are presented in Table 5. We find that denoising with grammatical error correction can not bring any improvement in terms of BLEU/chrF++/TER/BLEURT metrics. The proposed approach using ChatGPT as a monolingual denoising tool to revise the ill-generated translation output can achieve significant improvement over the baseline mBART-Large-50. We have not clearly known how the ChatGPT revises the ill-generated translation output and leave the investigation to future work.

[7] https://huggingface.co/pszemraj/flan-t5-large-grammar-synthesis.

[8] https://huggingface.co/vennify/t5-base-grammar-correction.

Table 6. Num of *cor* influence for model. The prefix in the name is the num of *cor*.

Metric	base	10W-enc	20W-enc	30W-enc
BLEU	23.40	23.60	23.75	**23.88**
chrF++	42.78	43.03	43.22	**43.48**

Data Size. We investigate the impact of data size on error-corrected data. After the identification and denoising process, we randomly select a certain number of training samples to finetune the mBART-Large-50 model. We present the translation results in Table 6. We find that the model exhibits a correspondingly enhanced performance as the quantity of data increases.

5 Conclusion

In this study, we propose a methodology to enhance pseudo-parallel data by employing monolingual denoising techniques. We leverage Language Models (LLMs) to enhance the performance of machine translation systems in low-resource settings. Experimental results demonstrate that applying the method to the low-resource machine translation model improves translation in terms of BLEU, chrF++, and TER metrics. For future work, we intend to examine the impact of LLMs on the experimental process and explore the extension of this method to other low-resource NLP tasks.

Acknowledgements. The research is supported by National Key R&D Program of China (2020YFB1313601), National Science Foundation of China (62076174, 61836007).

References

1. Aharoni, R., Johnson, M., Firat, O.: Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3874–3884. Minneapolis, Minnesota (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
3. Guerreiro, N.M., Voita, E., Martins, A.F.: Looking for a needle in a haystack: a comprehensive study of hallucinations in neural machine translation. arXiv preprint arXiv:2208.05309 (2022)
4. Haddow, B., Bawden, R., Barone, A.V.M., Helcl, J., Birch, A.: Survey of low-resource machine translation. Comput. Linguist. **48**(3), 673–732 (2022)
5. He, Z., et al.: Exploring human-like translation strategy with large language models. arXiv preprint arXiv:2305.04118 (2023)

6. He, Z., Wang, X., Tu, Z., Shi, S., Wang, R.: Tencent ai lab-shanghai jiao tong university low-resource translation system for the wmt22 translation task. In: Proceedings of the Seventh Conference on Machine Translation (WMT), pp. 260–267 (2022)
7. He, Z., Wang, X., Wang, R., Shi, S., Tu, Z.: Bridging the data gap between training and inference for unsupervised neural machine translation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6611–6623 (2022)
8. Jiao, W., Wang, W., Huang, J.t., Wang, X., Tu, Z.: Is chatgpt a good translator? a preliminary study. arXiv preprint arXiv:2301.08745 (2023)
9. Jiao, W., Wang, X., He, S., King, I., Lyu, M., Tu, Z.: Data rejuvenation: exploiting inactive training examples for neural machine translation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2255–2266 (2020)
10. Jiao, W., Wang, X., Tu, Z., Shi, S., Lyu, M., King, I.: Self-training sampling with monolingual data uncertainty for neural machine translation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2840–2850 (2021)
11. Khayrallah, H., Koehn, P.: On the impact of various types of noise on neural machine translation. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pp. 74–83 (2018)
12. Koehn, P., Khayrallah, H., Heafield, K., Forcada, M.L.: Findings of the wmt 2018 shared task on parallel corpus filtering. In: Proceedings of the third Conference on Machine Translation: Shared Task Papers, pp. 726–739 (2018)
13. Liang, T., et al.: Encouraging divergent thinking in large language models through multi-agent debate. arXiv preprint arXiv:2305.19118 (2023)
14. Lu, J., Ge, X., Shi, Y., Zhang, Y.: Alibaba submission to the wmt20 parallel corpus filtering task. In: Proceedings of the Fifth Conference on Machine Translation, pp. 979–984 (2020)
15. Nguyen, T.Q., Murray, K., Chiang, D.: Data augmentation by concatenation for low-resource translation: a mystery and a solution. In: IWSLT 2021, p. 287 (2021)
16. Nguyen, X.P., Joty, S., Wu, K., Aw, A.T.: Data diversification: a simple strategy for neural machine translation. Adv. Neural. Inf. Process. Syst. **33**, 10018–10029 (2020)
17. NLLB Team, Marta R. Costa-jussà, J.C.: No language left behind: Scaling human-centered machine translation (2022)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6–12, 2002, Philadelphia, PA, USA, pp. 311–318 (2002)
19. Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 392–395. Lisbon, Portugal (2015)
20. Popović, M.: chrF++: words helping character n-grams. In: Proceedings of the Second Conference on Machine Translation, pp. 612–618. Copenhagen, Denmark (2017)
21. Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191. Belgium, Brussels (2018)

22. Ranathunga, S., Lee, E.S.A., Prifti Skenduli, M., Shekhar, R., Alam, M., Kaur, R.: Neural machine translation for low-resource languages: a survey. ACM Comput. Surv. **55**(11), 1–37 (2023)
23. Sellam, T., Das, D., Parikh, A.: BLEURT: learning robust metrics for text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7881–7892. Online (2020)
24. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: 54th Annual Meeting of the Association for Computational Linguistics, pp. 86–96 (2016)
25. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pp. 223–231. Cambridge, Massachusetts, USA (2006)
26. Tang, Y., Tran, C., Li, X., Chen, P.J., Goyal, N., Chaudhary, V., Gu, J., Fan, A.: Multilingual translation with extensible multilingual pretraining and finetuning (2020)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
28. Vyas, Y., Niu, X., Carpuat, M.: Identifying semantic divergences in parallel text without annotations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1503–1515 (2018)
29. Wang, L., Lyu, C., Ji, T., Zhang, Z., Yu, D., Shi, S., Tu, Z.: Document-level machine translation with large language models. arXiv preprint arXiv:2304.02210 (2023)
30. Wang, R., Tan, X., Luo, R., Qin, T., Liu, T.Y.: A survey on low-resource neural machine translation. arXiv preprint arXiv:2107.04239 (2021)
31. Wang, W., Watanabe, T., Hughes, M., Nakagawa, T., Chelba, C.: Denoising neural machine translation training with trusted data and online data selection. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 133–143 (2018)
32. Wang, W., et al.: Understanding and improving sequence-to-sequence pretraining for neural machine translation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2591–2600 (2022)
33. Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., Zhang, M.: Neural machine translation advised by statistical machine translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
34. Wang, X., Pham, H., Dai, Z., Neubig, G.: Switchout: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 856–861 (2018)
35. Wu, H., Wang, W., Wan, Y., Jiao, W., Lyu, M.: Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. arXiv preprint arXiv:2303.13648 (2023)
36. Xia, M., Kong, X., Anastasopoulos, A., Neubig, G.: Generalized data augmentation for low-resource translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5786–5796. Florence, Italy (2019)
37. Xing, X., Hong, Y., Xu, M., Yao, J., Zhou, G.: Taking actions separately: a bidirectionally-adaptive transfer learning method for low-resource neural machine translation. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 4481–4491 (2022)

38. Zhang, B., Williams, P., Titov, I., Sennrich, R.: Improving massively multilingual neural machine translation and zero-shot translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1628–1639. Online (2020)
39. Zhang, J., Zong, C.: Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545 (2016)
40. Zoph, B., Yuret, D., May, J., Knight, K.: Transfer learning for low-resource neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1568–1575. Austin, Texas (2016)

Oral: Multimodality and Explainability

CAMG: Context-Aware Moment Graph Network for Multimodal Temporal Activity Localization via Language

Yuelin Hu[1], Yuanwu Xu[1], Yuejie Zhang[1](✉), Rui Feng[1], Tao Zhang[2], Xuequan Lu[3], and Shang Gao[3]

[1] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Shanghai Collaborative Innovation Center of Intelligent Visual Computing, Fudan University, Shanghai 200433, People's Republic of China
{19210240193,202102400093,yjzhang,fengrui}@fudan.edu.cn

[2] School of Information Management and Engineering, Shanghai Key Laboratory of Financial Information Technology, Shanghai University of Finance and Economics, Shanghai 200433, People's Republic of China
taozhang@mail.shufe.edu.cn

[3] School of Information Technology, Deakin University, Waurn Ponds, VIC 3216, Australia
{xuequan.lu,shang}@deakin.edu.au

Abstract. Temporal Activity Localization via Language (TALL) is a challenging task for language based video understanding, especially when a video contains multiple moments of interest and the language query has words describing complex context dependencies between the moments. Latest studies have proposed various ways to exploit the temporal context of adjacent moments, but two apparent limitations remained. First, only limited context information was encoded based on RNNs or 2-D convolutions, which highly depended on the pre-sorting of proposals and lacked flexibility. Second, semantically correlated content in different moments was ignored, i.e., semantic context. To address these limitations, we propose a novel GCN-based framework, i.e., Context-Aware Moment Graph (CAMG) network, to jointly model the temporal context and semantic context. Also, we design a multi-step fusion scheme to aggregate object, motion and textual features. A Query-Gated Integration Module is further designed to select queried objects and filter out noisy ones. Our model achieves superior performance to state-of-the-art methods on two widely-used benchmark datasets.

Keywords: Context-aware moment graph (CAMG) network · multimodal fusion · temporal activity localization via language · query-gated integration module

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-44693-1_34.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 429–441, 2023.
https://doi.org/10.1007/978-3-031-44693-1_34

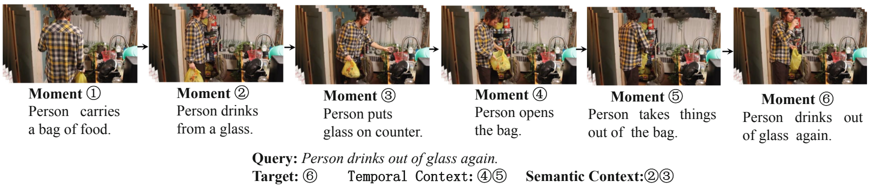

Moment ①
Person carries a bag of food.
Moment ②
Person drinks from a glass.
Moment ③
Person puts glass on counter.
Moment ④
Person opens the bag.
Moment ⑤
Person takes things out of the bag.
Moment ⑥
Person drinks out of glass again.
Query: Person drinks out of glass again.
Target: ⑥ Temporal Context: ④⑤ Semantic Context:②③

Fig. 1. An illustration of temporal context and semantic context.

1 Introduction

Temporal Activity Localization via Language (TALL) is a fundamental problem in language based video understanding. It aims to localize a moment in an untrimmed video that best matches a given description. This is a challenging task, especially when a video contains multiple moments of interest and the language query has words describing complex context dependencies between the moments, such as "*His face is getting toweled off* ***after*** *being shaven*".

Most previous work [5,8,9] generated candidate moments (i.e., proposals) via sliding window. These methods treated the surrounding frames as temporal context and incorporated them by extending the proposal boundaries. This strategy may fail dramatically when the query refers to the moments appearing outside of the extended proposals. Therefore, recent studies have proposed various ways to exploit temporal dependencies across proposals, e.g., utilizing 2-D convolutions to encode context from adjacent proposals [21] or gradually aggregating context through a Recurrent Neural Network (RNN) [22]. These methods have achieved good performance but still have two apparent limitations.

First, these methods typically rely on RNNs or 2D convolutions with a predefined kernel size, which may experience difficulties when encoding long-range temporal dependencies across proposals. In addition, their proposal features need to be organized chronologically for subsequent modules to process. E.g., to organize the proposal features, Zhang *et al.* [21] utilized a two-dimensional time map, which lacked flexibility to some extent.

Second, they only utilized the temporal context surrounding the target activity within a certain temporal distance. However, real-world videos may vary dramatically in content. Coupled with the complexity of language queries, the temporal context alone could not fully exploit the rich information suggested by the video context. As shown in Fig. 1, given a query like "*person drinks out of glass* ***again***", the moments preceding the target activity are "*opening a bag*" and "*taking things out*", which obviously do not help with the localization of the queried activity. However, the moments far away from the target activity, such as "*person drinking from a glass*" and "*person puts glass on counter*", are able to provide more indicative hints. We refer to such context as semantic context, which is semantically correlated to the target activity other than those located in its vicinity. This kind of semantic information is more desirable but has not attracted enough attention in the TALL task.

In this paper, we design a Context-Aware Moment Graph (CAMG) network based on Graph Convolutional Networks (GCNs) to address the aforementioned limitations for the TALL task. It models the temporal context and semantic context jointly through two sub-graphs, i.e., Temporal Moment Graph (TMG) and Semantic Moment Graph (SMG), both of which formulate candidate moments as graph nodes.

TMG perceives temporal context from adjacent moments within a predefined distance, where the edges encode the temporal relations between video moments. SMG perceives semantic context from semantically correlated moments. Due to the ever-changing video contents, it is hard to manually predefine the semantic neighbors for each moment Therefore, we utilize dynamic edge convolutions [16] to dynamically construct semantic edges between proposals. The semantic neighbors are dynamically selected from the entire video according to their content similarities to the target moment, making the SMG content-adaptive.

In addition, we observe that object-level feature is less exploited than motion feature in previous methods. Only a few works like [6] aggregated both object feature and motion feature by simple feature concatenation. In this work, we design a more powerful multi-step fusion scheme to fuse object, motion and textual feature step by step. Considering multiple objects may be contained in one frame, we design a Query-Gated Integration Module (QGIM) to further select queried objects and filter out noisy ones.

In summary, our contributions are as follows.

1. We propose a novel GCN-based framework, i.e., Context-Aware Moment Graph (CAMG) network to model the temporal context and semantic context jointly.
2. We design a powerful multi-step fusion scheme to aggregate object, motion and textual features. A Query-Gated Integration Module is further designed to select queried objects and filter out noisy ones.
3. Quantitative and qualitative experiments on two public benchmark datasets (i.e., Charades-STA and ActivityNet Captions) demonstrates the effectiveness of our designs. The state-of-the-art scores are achieved as well compared to the current approaches.

2 Related Work

2.1 Temporal Activity Localization via Language

In the TALL task, context information is a critical cue to help effectively localize the described activities. Early *sliding-window-based* methods which generate candidate moments (i.e. proposals) via sliding-window strategy simply treat the surrounding frames as temporal context and incorporate them by extending the proposal boundaries. A major drawback is that the moments out of the extended proposals will not be observed.

RNN-based methods [22] utilize a RNN to gradually aggregate temporal contexts from frame representations and generate proposals using anchors centered at each frame. However, it is still hard for the *RNN-based* methods to encode long-range temporal dependencies. *2-D convolution based* methods [21] organize proposals as a 2-D temporal map, and adopt 2-D convolutional layers to aggregate temporal context from adjacent proposals. However, these frameworks can only encode limited context information with a pre-defined kernel size.

Different from these two kinds of methods, our CAMG is based on GCN and able to capture the long-range dependencies with a high level of flexibility. Moreover, all the aforementioned methods only focus on the temporal context, while our CAMG takes semantic context into account as well.

2.2 Object-Level Visual Features in TALL

Compared with motion features, object-level features preserve more detailed and discriminative visual information. In the TALL task, Chen and Jiang [2] built a graph to model the object-object interactions and object-text interactions jointly. However, motion features were not used in their work, which limited their capability. Jiang *et al.* [6] combined object-level feature, motion feature and textual feature to improve the localization accuracy, where the three types of features were fused by vector concatenation. Different from their work, we design a multi-step fusion scheme and first utilize a gated attention mechanism to better integrate the object features in the TALL task.

3 Methodology

Our CAMG network is illustrated in Fig. 2. Multimodal features are first extracted by the Multimodal Feature Extractor, then fused by the Multi-step Fusion Module. Next, moment proposals are generated based on the fusion results. A novel Context-Aware Moment Graph is used to model the context dependencies across the proposals.

3.1 Multimodal Feature Extractor

Motion and Textual Features. Given a video V, we first segment it into N small video clips, each of which consists of T frames ($T = 16$ in C3D). For each clip, its motion feature is extracted from the pre-trained C3D network [14]. The motion features can be represented as $M = \{m_i\}_{i=1}^{N}$. The word representations $S = \{s_i\}_{i=1}^{L_q}$ of a query are obtained using GloVe embeddings [11], where L_q is the number of words. We project both features into the same dimension d by two BiGRUs [3].

Object-Level Visual Features. For each given video V, we perform a fixed-interval sampling and obtain N frames. Then, we extract object-level features for each sampled frame $O = \{\{o_i^j\}_{j=1}^{L_o}\}_{i=1}^{N}$, where $i \in [1, N]$ is the frame index, $j \in [1, L_o]$ is the object index, and L_o is the number of object features.

To further select queried objects and filter out noisy ones, we design a Query-Gated Integration Module (QGIM), as illustrated in Fig. 3. When integrating

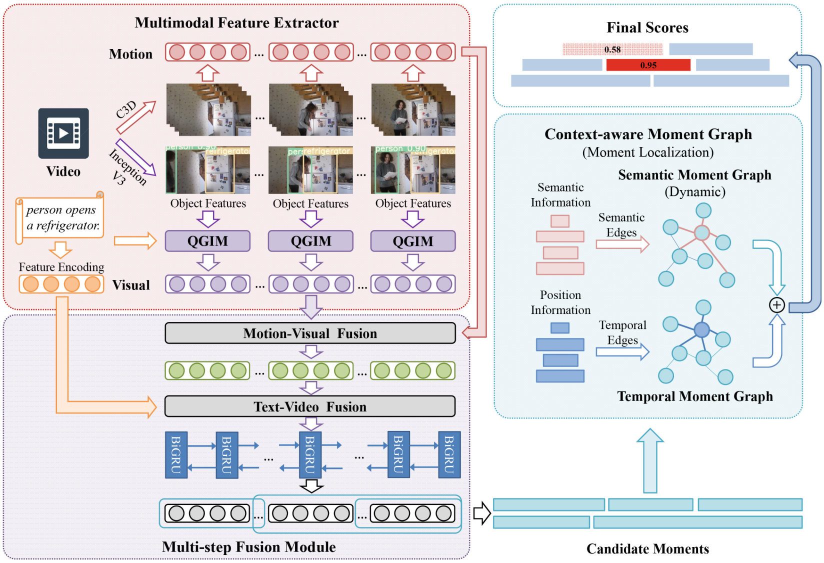


Fig. 2. The architecture of the proposed CAMG network. QGIM denotes the Query-Gated Integration Module.

the object features of each frame, channel-wise modulating gates are designed to adjust the object weights. For the i^{th} sampled frame, we first adopt a soft attention to and obtain the sentence feature f_s. Next, we create the modulating gates. We concatenate each pair of sentence feature f_s and object feature $o_i^j, j \in [1, L_o]$ as $[f_s; o_i^j]$, and use a fully connected (FC) layer W_g followed by a sigmoid activation σ to obtain the modulating gate G_i^j for each object.

$$G_i^j = \sigma(W_g[f_s; o_i^j]) \tag{1}$$

The object features are then modulated by these gates as follows:

$$\hat{o}_i^j = (1 + G_i^j) \odot o_i^j \tag{2}$$

where $\odot$ denotes element-wise multiplication.

Finally, we utilize a spatial attention to obtain a global representation $\overline{O} = \{\overline{o}_i\}_{i=1}^N$ for each frame. For the i^{th} sampled frame, the attention weight a_i^j assigned to the j^{th} object is computed from the modulated object feature $\hat{o}_i^j$, which is formulated as:

$$a_i^j = W_a^1 \tanh(W_a^2 \hat{o}_i^j) \tag{3}$$

where W_a^1 and W_a^2 are learnable parameters.

3.2 Multi-step Fusion Module

Given the extracted textual features S, the motion features M and the object-level features $\overline{O}$, these three types of features are fused by two steps. As illustrated in Fig. 2, we first fuse the motion features and object-level features to get a comprehensive representation of the video. We concatenate the motion features M and object-level features $\overline{O}$ as $[M; \overline{O}]$, and adopt a FC layer to obtain the joint feature vectors X.

As for textual information we first obtain a sentence representation $\overline{S}$ for each clip by soft attention. Then we concatenate the fused video features X and its corresponding sentence representation $\overline{S}$ as $[X; \overline{S}]$, and feed them into a BiGRU network with the output dimension of d. The final fusion result can be represented as $Y = \{y_i\}_{i=1}^{N}, y_i \in \mathbb{R}^d$. As for the generation of temporal proposals, we take the same sliding-window strategy as [5].

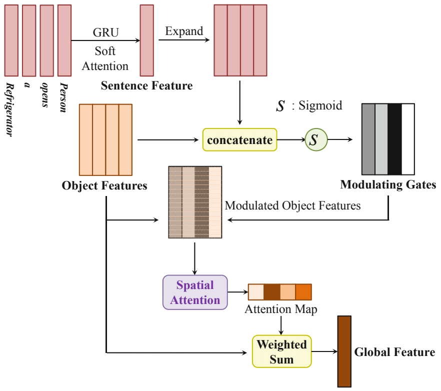


Fig. 3. The Query-Gated Integration Module (QGIM).

3.3 Context-Aware Moment Graph

Our CAMG models the temporal context and semantic context jointly. It consists of two sub-graphs, i.e., Temporal Moment Graph (TMG) and Semantic Moment Graph (SMG), both of which formulate moment proposals as graph nodes. We denote the node set as $H_0 = \{p_i\}_{i=1}^{L_p}$, where L_p is the total number of moment proposals.

Temporal Moment Graph. TMG aims to perceive the temporal context from immediate temporal neighbors. Therefore, we connect moment proposals p_i and p_j if $r(p_i, p_j) \geq \theta_{rel}$, where θ_{rel} is a pre-defined threshold, and $r(p_i, p_j)$ represents the temporal relevance defined by Intersection over Union (IoU) score. In order to exploit long-range temporal dependencies, non-overlapping proposals within a certain distance are also connected. For non-overlapping proposals (i.e., $r(p_i, p_j) = 0$), we compute the following distance between them:

$$d(p_i, p_j) = \frac{|c_i - c_j|}{L(p_i)} \tag{4}$$

where c_i and c_j are the center coordinates of p_i and p_j, respectively; and $L(p_i)$ represents the length of the target proposal p_i. If $d(p_i, p_j) < \theta_{dis}$, p_j will be connected to p_i, where θ_{dis} is the pre-defined threshold.

Unlike video frames, moments last for a certain period of time and are harder to be organized chronologically. To address this problem, we encode the temporal relations as the edges of TMG. The weight assigned to each edge $E_{Tem}(p_i, p_j)$ is defined as:

$$W_E(p_i, p_j) = \phi(Loc(p_i))^T \phi(Loc(p_j)) \tag{5}$$

where $Loc(p)$ denotes the location feature of proposal p, and ϕ represents a linear transformation. The location feature $Loc(p)$ is set to the concatenation of p_s and p_e, where p_s and p_e is the positional encoding [15] of the start and end clip indexes, respectively. Once computed for each edge, the edge weights are normalized and an identity matrix representing self-loop edges is added to obtain the final weight matrix $G_0 \in \mathbb{R}^{L_p \times L_p}$.

Finally, we apply a GCN on TMG. The update process is formulated as:

$$H_1^{Tem} = \tanh(G_0 H_0 W_0) \tag{6}$$

where $H_0 \in \mathbb{R}^{L_p \times d}$ denotes the input node features; $H_1^{Tem} \in \mathbb{R}^{L_p \times d}$ is the updated node representation; and $W_0 \in \mathbb{R}^{L_p \times L_p}$ is the learnable parameter.

Semantic Moment Graph SMG aims to perceive the semantic context from semantically correlated moments. Due to the ever-changing video contents, it is hard to manually pre-define the semantic neighbors for each moment. Therefore, we dynamically establish edges between moment proposals according to their feature similarities. Following the dynamic edge convolutions proposed by [16], we construct a k-nearest neighbor (k-NN) graph for the graph nodes H_0. The edge set for each moment proposal p_i in H_0 is defined as $\{(p_i, p_{n_i(j)}) | j \in \{1, 2, \dots, k\}\}$, where $n_i(j)$ is the node index of the j^{th} nearest neighbor of the node p_i. The *edge features* are defined as:

$$E_{sem}(p_i, p_{n_i(j)}) = h_\theta(p_i, p_{n_i(j)}) \tag{7}$$

where $h_\theta : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the edge function with a set of learnable parameters θ. In our work, we concatenate the node feature p_i and its neighbor feature $p_{n_i(j)}$ as $[p_i; p_{n_i(j)}]$, and adopt a MLP to obtain the edge features as follows:

$$h_\theta(p_i, p_{n_i(j)}) = \text{MLP}([p_i; p_{n_i(j)}]) \tag{8}$$

Finally, we adopt the edge convolutions by applying a channel-wise symmetric aggregation operation (*max* in our work) on the edge features corresponding to the edges emanating from node p_i. The update process is formulated as:

$$p_i' = \max_{j \in [1,k]} E_{sem}(p_i, p_{n_i(j)}) \tag{9}$$

where p_i' is the updated node representation of p_i. The final node representation is represented as $H_1^{Sem} = \{p_i'\}_{i=1}^{L_p}, p_i' \in \mathbb{R}^d$.

Moment Localization. The final output H of CAMG is the aggregation of temporal and semantic sub-graphs, which can be formulated as:

$$H = H_1^{Tem} + H_1^{Sem}, H \in \mathbb{R}^{L_p \times d} \tag{10}$$

Then, we utilize the output feature H to predict the matching scores of all the candidate moments with the given query. The output feature H is passed through a FC layer followed by a sigmoid function, and L_p matching scores are obtained, denoted as $\hat{P} = \{\hat{p}_i\}_{i=1}^{L_p}$. Each value $\hat{p}_i$ represents the matching score between a candidate moment and the given query. The maximum value indicates the best matching moment. Our framework is trained by a binary cross-entropy loss:

$$Loss = \frac{1}{L_p} \sum_{i=1}^{L_p} y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \tag{11}$$

where y_i is the scaled IoU score of each candidate moment with the ground-truth moment.

4 Experiment and Analysis

4.1 Experimental Settings

Datasets. We conduct experiments on two popular benchmarks: Charade-STA [5] and ActivityNet Captions [7]. Charades-STA is prepared by Gao *et al.* [5], which contains 9,848 videos. There are 12,408 and 3,720 moment-query pairs in the training and testing set, respectively. ActivityNet Captions consists of 19,209 videos. Following [2,21], we split 37,417, 17,505 and 17,031 moment-query pairs for training, validation and testing, respectively.

Evaluation Metric. Following Gao *et al.* [5], we utilize "R@n, IoU=m" to evaluate our model. It denotes the percentage of language queries having at least one matched moment (the IoU score with the ground-truth moment is greater than m) in the top-n retrieved moments. Following the standard practice, we use $n = 1$ and $m \in \{0.3, 0.5, 0.7\}$.

Implementation Details. The number of sampled clips N is set to 30 for Charades-STA and 200 for ActivityNet Captions. For a fair comparison, we adopt the same motion features (e.g., C3D [14] features) as most of previous work. We use YOLO-v5[1] to extract bounding boxes for each sampled frame and adopt Inception-V3 [13] to get the visual representation of each box. Inception-ResNet-V2 [12] feature is also adopted for a fair comparison. The feature dimension d is set to 512. As for CAMG, θ_{rel} is set to 0.3 on Charades-STA and 0.5 on ActivityNet Captions, θ_{dis} is set to 2 on Charades-STA and 1.5 on ActivityNet Captions, and k is set to 4 on both datasets. We utilize Adam as the optimizer, and set the learning rate of 4×10^{-4} and 8×10^{-4} for Charades-STA and ActivityNet Captions, respectively.

4.2 Comparisons with the State-of-the-Art Methods

We compare our approach with the state-of-the-art TALL methods, including:

1. GCN-based methods: MAN [20].

[1] https://github.com/ultralytics/yolov5.

Table 1. Performance comparison on Charades-STA. “R”: ResNet-101; “IRV2”: Inception-ResNet-V2; “IV3”: Inception-V3; “A”: Audio feature; “Obj.”: Object-level feature; “✓”: enabled; “×”: disabled.

Method	Year	Feature	Obj.	R@1	R@1	R@1
				IoU=0.3	IoU=0.5	IoU=0.7
CTRL [5]	2017	C3D	×	-	23.63	8.89
ABLR [19]	2019	C3D	×	51.55	35.43	15.05
I^{2}N [10]	2021	C3D	×	-	*41.69*	*22.89*
BPNet [18]	2021	C3D	×	*55.64*	38.25	20.51
Ours	-	C3D	×	**62.10**	**48.33**	**26.53**
SLTA [6]	2019	C3D+R	✓	38.96	22.81	8.25
HVTG [2]	2020	IRV2	✓	61.37	47.27	23.30
PMI-LOC [1]	2020	C3D+A+IRV2	×	58.08	42.63	21.32
Ours	-	C3D+IV3	✓	**65.78**	**51.34**	*28.36*
Ours	-	C3D+IRV2	✓	*64.70*	*51.24*	**28.87**
MAN [20]	2019	VGG	×	-	*41.24*	20.54
2D-TAN [21]	2020	VGG	×	-	40.94	*22.85*
MMN [17]	2022	VGG	×	-	47.31	**27.28**
Which-Where [4]	2022	VGG	×	-	45.64	*26.13*
Ours	-	VGG	×	**63.66**	**48.28**	26.08

2. Sliding-window based methods: CTRL [5].
3. RNN-based and 2-D convolution based methods: CMIN [22], 2D-TAN [21], MMN [17] and Which-Where [4].
4. Multimodal and object-based methods: SLTA [6], HVTG [2] and PMI-LOC [1].
5. Other methods: ABLR [19], BPNet [18], I^{2}N [10] and DTR [23].

Results on Charades-STA. Table 1 shows the performance comparison on Charades-STA. The first place is in **bold** and the second is in *italic*. For a fair comparison, we utilize the same motion features as each compared method. Our CAMG network outperforms most of previous works and achieves comparable performance with the latest works. Compared to the methods only regarding temporal context modelling, i.e., CTRL and 2D-TAN, the advantages of our CAMG network are also remarkable. It verifies the effectiveness of joint modeling of the temporal context and semantic context. Compared to the multimodal and object-based methods, i.e., SLTA, HVTG and PMI-LOC, Our CAMG network outperforms all the three methods by aggregating the object, motion and textual features through the multi-step fusion scheme.

Results on ActivityNet Captions. Table 2 summarizes the results on ActivityNet Captions which has more diverse and open video contents. As shown in Table 2, our CAMG network outperforms most of previous works and achieves comparable performance with the latest works even with the C3D feature alone.

Table 2. Performance comparison on ActivityNet Captions.

Method	Year	Feature	Obj.	R@1 IoU=0.3	R@1 IoU=0.5	R@1 IoU=0.7
CTRL [5]	2017	C3D	×	47.43	29.01	10.34
ABLR [19]	2019	C3D	×	55.67	36.79	-
CMIN [22]	2019	C3D	×	63.61	43.40	23.88
2D-TAN [21]	2020	C3D	×	59.45	44.51	*26.54*
BPNet [18]	2021	C3D	×	58.98	42.07	24.69
DTR [23]	2022	C3D	×	**65.38**	**47.03**	26.12
Ours	-	C3D	×	*64.58*	*46.68*	**26.64**
HVTG [2]	2020	IRV2	✓	57.5	40.15	18.27
PMI-LOC [1]	2020	C3D+IRV2+A	×	61.22	40.07	18.29
Ours	-	C3D+IV3	✓	**65.93**	*47.13*	*27.11*
Ours	-	C3D+IRV2	✓	*64.90*	**47.27**	**27.24**

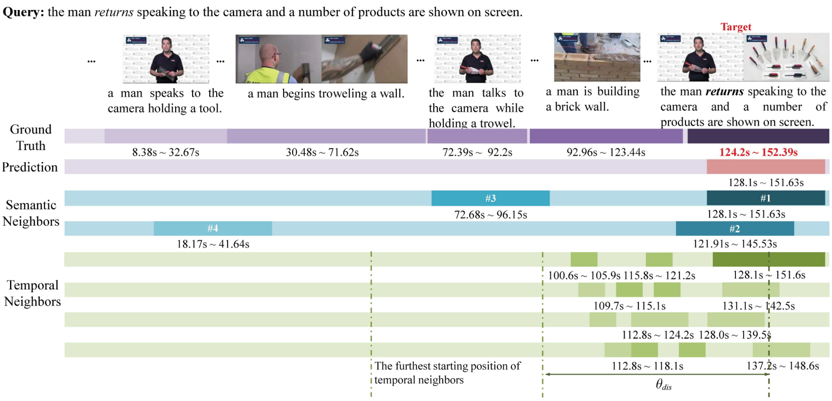

Query: the man returns speaking to the camera and a number of products are shown on screen.
Target
a man speaks to the camera holding a tool.
a man begins troweling a wall.
the man talks to the camera while holding a trowel.
a man is building a brick wall.
the man returns speaking to the camera and a number of products are shown on screen.
Ground Truth
8.38s ~ 32.67s
30.48s ~ 71.62s
72.39s ~ 92.2s
92.96s ~ 123.44s
124.2s ~ 152.39s
Prediction
128.1s ~ 151.63s
Semantic Neighbors
#3
#1
72.68s ~ 96.15s
128.1s ~ 151.63s
#4
#2
18.17s ~ 41.64s
121.91s ~ 145.53s
Temporal Neighbors
100.6s ~ 105.9s
115.8s ~ 121.2s
128.1s ~ 151.6s
109.7s ~ 115.1s
131.1s ~ 142.5s
112.8s ~ 124.2s
128.0s ~ 139.5s
112.8s ~ 118.1s
137.2s ~ 148.6s
The furthest starting position of temporal neighbors
θ_{dis}

Fig. 4. The visualization of temporal and semantic neighbors.

When compared to the methods utilizing multimodal features and object-level features, i.e., HVTG and PMI-LOC, our CAMG network achieves particularly superior results on all evaluation metrics, while the performance gain brought by the multimodal features of our CAMG network is not as obvious as that on Charades-STA. This may be due to the more complex and diverse video contents of ActivityNet Captions.

4.3 Qualitative Results

In Fig. 4, we visualize the two types of neighbors chosen by our CAMG model. Note that we only visualize the temporal neighbors with higher edge weights (>0.015). Semantic neighbors are selected based on content similarity as it is able to ignore the temporal distance. As illustrated in Fig. 4, the target moment is mainly about "*The man* ***returns*** *speaking to the camera*". It is the third time that the man talks to the camera throughout the whole video. The fourth and the third nearest semantic neighbors accurately contain the first and the second talk, respectively, both of which have high semantic similarities with the target moment. These two neighbors indicate that other occurrences of the target activity exist in previous moments, which provides a key clue to the localization of the queried activity.

4.4 Ablation Study

We examine the effectiveness of the multimodal feature, QGIM, and the two sub-graphs (i.e., TMG and SMG). The results are presented in Table 3.

First, the experiments are split into two groups, i.e., with the C3D feature only and with multimodal feature. It can be observed that the object-level feature has significant impact on the overall performance, as the feature is able to provide extra detailed and discriminative information helping the model achieve a better video understanding.

Table 3. Module analysis on both datasets. "Q": QGIM, Query-Gated Integration Module; "T": TMG, Temporal Moment Graph; "S": SMG, Semantic Moment Graph; "Δ": Improvements against the baseline model in each group.

Row	Feature	Q	CAMG		Charades-STA		ActivityNet Captions	
			T	S	R@1 IoU = 0.5	R@1 IoU = 0.7	R@1 IoU = 0.5	R@1 IoU = 0.7
					(Δ)	(Δ)	(Δ)	(Δ)
1	C3D	–	×	×	44.61 (+0.00)	23.06 (+0.00)	44.92 (+0.00)	24.61 (+0.00)
2	C3D	–	✓	×	48.01 (+3.40)	25.89 (+2.83)	46.42 (+1.50)	25.98 (+1.37)
3	C3D	–	×	✓	47.90 (+3.29)	25.91 (+2.85)	45.73 (+0.81)	25.69 (+1.08)
4	C3D	–	✓	✓	48.33 (+3.72)	26.53 (+3.47)	46.68 (+1.76)	26.64 (+2.03)
5	C3D+IV3	×	×	×	47.20 (+0.00)	25.67 (+0.00)	45.69 (+0.00)	25.84 (+0.00)
6	C3D+IV3	✓	×	×	48.95 (+1.75)	25.99 (+0.32)	46.30 (+0.61)	25.88 (+0.04)
7	C3D+IV3	✓	✓	×	49.49 (+2.29)	26.94 (+1.27)	46.76 (+1.07)	26.36 (+0.52)
8	C3D+IV3	✓	×	✓	50.35 (+3.15)	26.75 (+1.08)	47.13 (+1.44)	26.70 (+0.86)
9	C3D+IV3	✓	✓	✓	51.34 (+4.14)	28.36 (+2.69)	47.13 (+1.44)	27.11 (+1.27)
10	C3D+IV3	×	✓	×	49.78 (+2.58)	25.81 (+0.14)	46.45 (+0.76)	26.39 (+0.55)
11	C3D+IV3	×	×	✓	50.13 (+2.93)	26.21 (+0.54)	46.75 (+1.06)	25.96 (+0.12)

Next, we focus on the effect of QGIM. In Row 5, we remove the QGIM by computing the spatial attention weights on the original object features instead of the modulated ones. Comparing Row 6 and Row 5, it can be observed that QGIM can effectively improve the performance. The average improvements over all metrics on Charades-STA and ActivityNet are 1.05% and 0.41%, respectively.

Finally, we focus on the effects of TMG and SMG. As shown in Table 3, both the sub-graphs provide great performance gain. The average improvements over the three metrics with the C3D feature alone by TMG (Row 1 vs. Row 2) and SMG (Row 1 vs. Row 3) are 2.36% and 2.23% On Charades-STA, and 1.16% and 0.93% on ActivityNet Captions. It indicates that our CAMG can benefit from both types of context information in locating the target activities. Moreover, combining the two sub-graphs yields better performance consistently. It reveals that the integration of temporal and semantic context is able to provide a more comprehensive and complementary understanding of the queried video.

5 Conclusion

To fully exploit the context information for the TALL task, we propose a novel GCN-based model (i.e., Context-Aware Moment Graph) to model the temporal context and semantic context jointly. The aggregation of temporal and semantic context is able to provide a comprehensive and complementary understanding of the queried video. Moreover, a multi-step fusion scheme is designed to aggregate multimodal features, together with a Query-Gated Integration Module to select the queried objects. In the future, we would like to exploit more modalities such as audio for a better feature representation.

Acknowledgment. This work was supported by National Science and Technology Innovation 2030 - Major Project (No. 2021ZD0114001; No. 2021ZD0114000), National Natural Science Foundation of China (No. 61976057; No. 62172101), and the Science and Technology Commission of Shanghai Municipality (No. 21511101000; No. 22DZ1100101).

References

1. Chen, S., Jiang, W., Liu, W., Jiang, Y.-G.: Learning modality interaction for temporal sentence localization and event captioning in videos. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 333–351. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_20
2. Chen, S., Jiang, Y.-G.: Hierarchical visual-textual graph for temporal activity localization via language. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12365, pp. 601–618. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58565-5_36
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
4. Gao, J., Sun, X., Ghanem, B., Zhou, X., Ge, S.: Efficient video grounding with which-where reading comprehension. TCSVT **32**, 6900–6913 (2022)
5. Gao, J., Sun, C., Yang, Z., Nevatia, R.: TALL: temporal activity localization via language query. In: ICCV, pp. 5267–5275 (2017)
6. Jiang, B., Huang, X., Yang, C., Yuan, J.: Cross-modal video moment retrieval with spatial and language-temporal attention. In: ICMR, pp. 217–225 (2019)
7. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: ICCV, pp. 706–715 (2017)
8. Liu, M., Wang, X., Nie, L., He, X., Chen, B., Chua, T.S.: Attentive moment retrieval in videos. In: ACM SIGIR, pp. 15–24 (2018)

9. Liu, M., Wang, X., Nie, L., Tian, Q., Chen, B., Chua, T.S.: Cross-modal moment localization in videos. In: ACM MM, pp. 843–851 (2018)
10. Ning, K., Xie, L., Liu, J., Wu, F., Tian, Q.: Interaction-integrated network for natural language moment localization. TIP **30**, 2538–2548 (2021)
11. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
12. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: AAAI (2017)
13. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 (2016)
14. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV, pp. 4489–4497 (2015)
15. Vaswani, A., et al.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
16. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. Acm Trans. Graph. (TOG) **38**(5), 1–12 (2019)
17. Wang, Z., Wang, L., Wu, T., Li, T., Wu, G.: Negative sample matters: a renaissance of metric learning for temporal grounding. In: AAAI, vol. 36, pp. 2613–2623 (2022)
18. Xiao, S., et al.: Boundary proposal network for two-stage natural language video localization. arXiv preprint arXiv:2103.08109 (2021)
19. Yuan, Y., Mei, T., Zhu, W.: To find where you talk: temporal sentence localization in video with attention based location regression. In: AAAI, vol. 33, pp. 9159–9166 (2019)
20. Zhang, D., Dai, X., Wang, X., Wang, Y.F., Davis, L.S.: MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. In: CVPR, pp. 1247–1257 (2019)
21. Zhang, S., Peng, H., Fu, J., Luo, J.: Learning 2D temporal adjacent networks for moment localization with natural language. In: AAAI, vol. 34, pp. 12870–12877 (2020)
22. Zhang, Z., Lin, Z., Zhao, Z., Xiao, Z.: Cross-modal interaction networks for query-based moment retrieval in videos. In: ACM SIGIR, pp. 655–664 (2019)
23. Zhou, H., Zhang, C., Luo, Y., Hu, C., Zhang, W.: Thinking inside uncertainty: interest moment perception for diverse temporal grounding. TCSVT **32**, 7190–7203 (2022)

Semantic Extension for Cross-Modal Retrieval of Medical Image-Diagnosis Report

Guohui Ding, Qi Zhang(✉), Shizhan Geng, and Chunlong Fan

Shenyang Aerospace University, Shenyang, China
{dingguohui,FanCHL}@sau.edu.cn, m17351291806@163.com

Abstract. In recent years, cross-modal hash retrieval has provided a new perspective for computer-aided diagnostic systems. However, in the field of cross-modal retrieval, the current mainstream methods all adopt a simple similarity discrimination strategy based on shared labels or the number of shared labels, which will lead to the isolation of semantics between labels and the loss of similar retrieval results. In this study, we propose a cross-modal retrieval method called Semantic Extension for Cross-Modal Retrieval of medical images and diagnostic reports (SECMR) designed to exploit hierarchal semantic associations between labels to constrain a hash code with more semantics. By mining the hierarchical association between disease labels and transforming it into multi-level semantic matrices, different weights are assigned according to the granularity level to guide the learning of image and text hash codes. In addition, we also introduce the focal loss function to mitigate class imbalance, which is a common problem with classification models based on medical data. Experiments conducted on the MIMIC-CXR dataset demonstrate that our cross-modal retrieval method surpasses other classic cross-modal retrieval models and achieves impressive performance.

Keywords: Medical cross-modal retrieval · Semantic Extension · Deep learning

1 Introduction

With the rapid application of computing and big data technologies in medicine and the gradual improvement of medical information storage standards, the quantity of multi-modal medical data continues to increase at an accelerating rate. Computer-aided diagnosis systems that only involve a single modality are largely not well-suited for clinical needs, thus medical cross-modal retrieval methods have been developed. Medical cross-modal retrieval refers to the use of instances of a given modality such as X-ray image to retrieve instances of another modality such as diagnostic reports, which helps doctors retrieve more comprehensive information to improve the efficiency and accuracy of clinical diagnosis. In addition, these integrated medical resources can also be used as learning materials for novice doctors and intern students, providing a new perspective for promoting computer-aided diagnosis. Therefore, research on medical cross-modal retrieval has important clinical significance in the medical field.

In the field of cross-modal retrieval, the current mainstream methods all adopt a simple similarity discrimination strategy based on shared labels or the number of shared

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 442–455, 2023.
https://doi.org/10.1007/978-3-031-44693-1_35

labels, which will lead to the isolation of semantics between labels and the loss of similar retrieval results. The underlying hierarchal association between disease labels of medical images and text is typically inherent and explicit, and thus implies a rich semantics. Hierarchal associations can be used to extend queries with higher hierarchical semantics to retrieve more comprehensive results. In this way, for some intractable diseases or rare diseases, the method will return similar cases through semantic expansion to provide some suggestions and ideas for doctors instead of returning no result. Thus, by gradually extending query semantics to higher-level concepts, the semantic structure of the original space can be maintained more effectively to support the retrieval of more comprehensive relevant information and improve performance on cross-modal retrieval tasks.

To the best of our knowledge, no methods have been reported that use hierarchal associations to improve the performance of cross-modal retrieval in the medical field. In this study, we propose a novel cross-modal hash retrieval framework based on underlying hierarchal associations between disease labels called Semantic Extension for Cross-Modal Retrieval of medical images and diagnostic reports (SECMR). To ensure that the hash code accurately preserves the semantic structure of the original space, we focus on two critical aspects. Firstly, we explore how to discover potential hierarchical semantic associations between labels. Secondly, we utilize this information to train a model that generates hash codes designed to maintain the original space's hierarchical structure. Word embeddings are capable of capturing the semantic information of words based on context. However, current pre-trained word embedding models [1] are not suitable for accurately embedding disease labels due to insufficient training data. To address this obstacle, we trained an unsupervised Word2Vec model based on a large medical text corpus, which maps disease labels into embedding vectors. Subsequently, we employ hierarchical clustering techniques to cluster the embedding vectors of disease labels into a hierarchy, thus obtaining hierarchical associations among diseases. With respect to the second aspect, we first used EfficientNetV2 and Sent2Vec models to extract features of medical images and diagnostic texts, respectively, while transforming the hierarchal associations into multi-level semantic matrices. Then, the matrices are assigned different weights based on their granularity level to supervise the process of learning to generate image and text hash codes. Supervising the training process with multilevel semantic matrices makes the hash codes more discriminative and allows them to reflect a rich semantic information. Machine learning models in the medical field are commonly subject to class imbalance, which causes the training process to fail to capture the semantic features of categories with relatively few samples. We introduce a focal loss function to optimize the proposed SECMR model to pay more attention to categories with few samples, which are otherwise difficult to identify. The contributions of this study are summarized as follows.

- We propose a cross-modal retrieval method for medical X-ray images and diagnostic reports. To the best of our knowledge, this is the first attempt to employ hierarchical associations to improve the performance of cross-modal retrieval in the medical field.
- The underlying hierarchical associations between labels are mined by an unsupervised word-embedding model to serve as supervised information to learn a hash

code. Thus, SECMR obtains more relevant retrieval results through hierarchical expansion of query semantics. In addition, we also introduce a focus loss function to mitigate class imbalance, which is a common problem with classification models based on medical data.
- We conducted extensive experiments on the public MIMIC-CXR datasets, and the results show that SECMR outperformed state-of-the-art cross-modal hashing methods.

The rest of this paper is organized as follows. Related work in cross-modal hashing is introduced in Sect. 2. We present our SECMR method in Sect. 3. Optimization algorithm is introduced in Sect. 4. Experiments are shown in Sect. 5. At last, we conclude our work in Sect. 6.

2 Related Works

Researches in cross-modal retrieval falls into two categories of supervised and unsupervised methods. Unsupervised methods [2,3,12,19] learn relations between samples with different modalities from unlabeled training data. For example, CMFH [2]learns the unified hash code of heterogeneous data by collective matrix factorization. To overcome the limitations of linear projection, LLSH [19] model sparse coding and matrix factorization are used to capture potential high-level semantic information in images and texts. Compared with supervised methods, although the unsupervised methods above are less data-constrained and have the advantages of saving labor in sorting out the massive data labeling process, their performance is relatively poor.

Supervised methods [4–7,10,14,14,17,18,20] use label information to guide the learning process of different modal data hash codes, and the rich semantic information of labels improves the performance of the model greatly. For example, Classical Correlation Analysis (CCA) [10] regards text features and visual features as different feature spaces, and learns the shadow space by maximizing the correlation of projection vectors. Semantic correlation maximization (SCM) [15] constructs a semantic similarity matrix with supervision information, embedding label information into hash codes of heterogeneous data. It is worth noting that the learning of feature extraction and hash code in these two methods is two-stage, which cannot keep the consistency of hash code and feature extraction well. To solve this problem, Deep Cross-Modal Hashing (DCMH) [4] unifies feature extraction and hash code learning in the same framework for the first time, achieving the end-to-end learning. Self-supervised adversarial hashing (SSAH) [5] is an early attempt to incorporate adversarial learning into cross-modal hashing in a self-supervised way. It uses two adversarial networks to maximize the semantic correlation and representation consistency between different modalities. In order to supervise the feature extraction model to pay more attention to the discriminative features, DSCMR [17] minimizes the discriminative loss in the label space and the common space. Moreover, it simultaneously minimizes the modality invariance loss and eliminates the cross-modal Representational differences of data to learn modality-invariant features.

Although these methods have achieved great success in cross-modal retrieval performance, it is inappropriate to treat all labels as isolated and semantically equidistant from each other because there are also potential semantic correlations between different labels. Different from existing cross-modal hashing methods, our work employs Word2vec model to mine latent semantic relationships between data and utilizes multi-level semantic as the supervision information for hash code learning, thus both semantic and structure similarities are well preserved, leading to optimal hash codes and better retrieval performance.

3 Proposed SECMR

In this section, we present the details about our SECMR method. The overall framework of the proposed method is shown in Fig. 1. We use a two-channel neural network to extract features from x-ray images and diagnosis reports, respectively. In addition, the similarity is not determined solely by the number of shared labels. Except for the fine-grained supervision information of shared labels, latent dependencies mined in a large number of medical corpora also guide the neural network to learn the discriminative regions. So the rich semantic information in the data is preserved in the common representation of the samples. The detailed introduction is as follows.

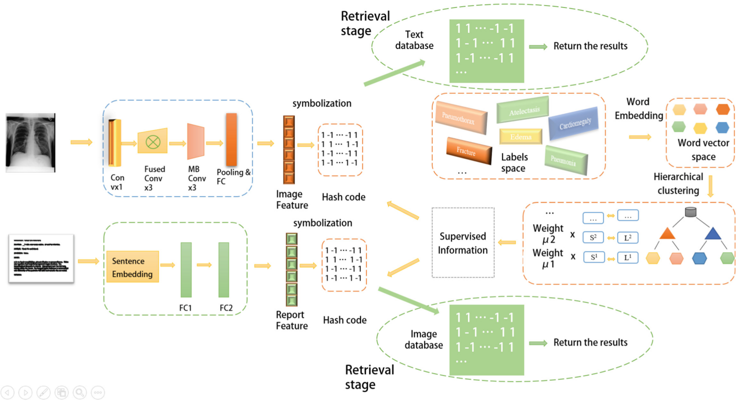


Fig. 1. The framework of our proposed method SECMR model

3.1 Problem Definition

Without losing generality, the bimodal case of cross-modal retrieval is only considered in the paper, i.e., for image and text. Firstly, the formal definition of bimodal data is presented as follows. Given a cross-modal training dataset with N instances, i.e., we use X = $\{x_i\}_{i=1}^n$ to denote the original x-rays images and use Y = $\{y_i\}_{i=1}^n$ to denote the diagnosis reports. Where y_i contains common semantic information related to x_i. The first-level label matrix is defined according to the original label set of the sample, which is denoted as $L^1 = \left\{l_{i1}^1, l_{i2}^1, \ldots, l_{ic}^1\right\} \in \{0, 1\}$, where c is the number of labels of the dataset. If the i-th sample belongs to the j-th categories, $l_{ij}^1 = 1$, otherwise $l_{ij}^1 = 0$. We also give a first-level similarity matrix S^1. $S_{ij}^1 = 1$ while x-rays image and diagnosis report at least one label is shared, otherwise $S_{ij}^{1} = 0$. As shown in Fig. 1, by performing hierarchical clustering on word embedding vectors of disease labels, we can obtain the multi-level semantic labels. In this article, we take the second-level semantics as an example to verify, and define the second-level label as $L^2 = \left\{ l_{i1}^2, l_{i2}^2 \ldots, l_{im}^2 \right\} \in \{0, 1\}$, where m is the number of clusters in the clustering algorithm. If the label of the i-th sample is in the k-th cluster, then $l_{ik}^2 = 1$, otherwise $l_{ik}^2 = 0$. In the same way, the secondary similarity matrix S^2 is obtained according to whether the label of samples are in the same cluster. $S_{ij}^{2} = 1$ while the label of x-rays image and diagnosis report in the same cluster, otherwise $S_{ij}^{2} = 0$.

Our goal is to learn hash representations of these two modalities: $B^{(x)} \in \{-1, +1\}^k$ for the modality of images and $B^{(y)} \in \{-1, +1\}^k$ for the modality of text, where k is the length of binary code. These Hamming distances between hash codes should preserve the cross-modal similarity in S^1 and S^2. More specifically, if $S_{ij}^{1} = 1$, the Hamming distance between $b_i^{(x)}$ and $b_j^{(y)}$ should be smaller than $S_{ij}^{2} = 1$, and if $S_{ij}^{2} = 0$, the Hamming distance should be maximum.

3.2 Feature Learning

The feature learning contains two deep neural networks, one for image modality and the other for text modality. Most medical images are single-channel gray images, and because human tissues and bones are very similar, their visual differences are small and concentrated in the lesion area. Therefore, we use EfficientNetV2 [13] to extract features of medical images, which obtains more fine-grained feature information by simultaneously increasing the depth and width of the network. In addition, EfficientNetV2 also increases the input image resolution, which is beneficial for extracting more detailed and comprehensive lesion information. In order to balance the accuracy of information extraction and the training time of the network, MBConv [11] and fused MBConv [13] modules are introduced into the EfficientNetV2 network structure, which greatly simplifies the network. In this way, the number of parameters and training time can be reduced while obtaining high-fine-grained features. The detailed configuration of EfficientNetV2 is shown in Table 1.

Table 1. Configuration of the network for image modality

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1,k3x3	1	24	2
2	Fused-MBConv4,k3x3	2	48	4
3	Fused-MBConv4,k3x3	2	64	4
4	MBConv4,k3x3,SE0.25	2	128	6
5	MBConv6,k3x3,SE0.25	1	160	9
6	MBConv6,k3x3,SE0.25	2	272	15
7	Conv1x1&Pooling&FC	–	1280	1
8	FC	–	hash length k	1

For the image modality, an image input x_i is resized to 300 × 380 and then fed into the cross-modal subnetwork to extract the high-level semantic features. We can denote the obtained modality specific feature vector for the i-th image x_i as: $f_i = \varphi(x_i; \theta_x) \in R^c, \theta_x$ is the network parameter of the EfficientNetV2 for image modality. In the existing cross-modal methods, the processing of text modalities generally uses the bag-of-words vector model or word embedding (i.e., word2vec).

The Bag-of-words vector will cause the matrix to be sparse, which seriously wastes memory and computing resources. In addition, it is based on individual words, without considering the order and connection between words in the sentence. Although the word embedding method solves the problem of sparse matrix, it still does not consider the order and connection between words in the sentence, so neither can well maintain the semantic structure of the sentence and fully express the meaning of the sentence. Different from the existing cross-modal methods, we use sentence embedding called Sent2vec, which makes two improvements in order to capture the semantic meaning of a sentence instead of being limited to a specific word: 1) The object of investigation is a whole sentence, rather than a sequence of words framed in a sentence with a fixed size window. 2) Introduce n-gram to enhance the embedding ability of word order in sentences [9]. It maps the entire sentence into a vector, rather than being limited to a single word, which preserves the semantic structure information in the sentence as more complete and comprehensive. For the text modality, the sentence vectors obtained from the Sent2vec modal are used as the input to a deep neural network with two fully-connected layers. The detailed configuration of a deep neural network for text is shown in Table 2.

Table 2. Configuration of the deep neural network for text modality

Layer	Configuration
Sentence embedding	700
FC1	512
FC2	hash length k

We can denote the obtained modality-specific feature vector for the j-th diagnostic report y_j as: $g_j = \psi\left(y_j; \theta_y\right) \in R^c, \theta_y$ is the network parameter of the neural network for text modality.

3.3 Multi-level Semantic Extension

The acquisition of multi-level semantic correlation is an essential content of the proposed method, and the primary problem to be solved is mining the potential correlation of samples from the dependencies of labels. First, we crawled a large number of medical texts from the Mayo Clinic website as the training corpus of the word embedding model. Skip-gram in Word2vec [8] is used as a word vector embedding model. The sample with a window size of 3 is selected as the positive sample of the central word, and 100 words are randomly selected as the negative sample in the full text. The objective function is defined as formula (1).

$$\sum_{w=1}^{W} \log \sigma\left(u_w^T V_c\right) + \sum_{k=1}^{K} \log \sigma\left(-\bar{u}_w{}^T V_c\right) \tag{1}$$

where $\sigma(\mathrm{x}) = 1/(1 + \exp(-\mathrm{x}))$, u_w and $\bar{u}_w$ represent positive and negative samples, respectively. W is the size of the training context, and K is the number of negative samples. In this article, we set $w = 3$ and $k = 100$. In this case, we obtained the embedded word vector with the original word semantics after many repeated iterations. After the word vector corresponding to the label is obtained from the trained word embedding model, potential correlation between the labels is outlined by hierarchical clustering algorithm, which serves as the coarse-grained similarity between the sample's correlation.

Another vital module is relevance ranking. To preserve semantic correlation between multi-modal data, we use pairwise correlation ranking constraints to make distance different between samples with different correlations according to the principle of more relevant the closer in the common semantic space. We construct multiple similarity matrices using semantic similarity between labels, and the fine-grained and coarse-grained similarities work together to learning of the hash code. In fact, the different similarity granularities can be obtained by adjusting the parameters in the clustering algorithm, and the number of similarity matrices is scalable. This paper designs a first-level similarity matrix S^1 and a second-level similarity matrix S^2, and the similarity matrices of different levels determine the pairwise ordering constraints by assigning different weights.

3.4 Hash-Code Learning

For the feature vectors of image modality and text modality obtained from the two-channel neural network, we use $F \in R^{k \times n}$, and $G \in R^{k \times n}$ denotes the learned feature representation matrix for the x-ray images and diagnosis reports, respectively. They are mapped into hash codes via sign function as shown in formula (2) and (3).

$$B^{(x)} = sign(F) \tag{2}$$

$$B^{(y)} = sign(G) \tag{3}$$

In order to keep the semantic structure information of the feature vector learned by the deep neural network, the loss function J1 defines as formula (4).

$$J1 = -\sum_{i,j=1}^{n} [((S_{ij}^{1}\theta_{ij} - \log(1 + e^{\theta_{ij}})) + \mu(S_{ij}^{2}\theta_{ij} - \log(1 + e^{\theta_{ij}}))] \tag{4}$$

The term $J1$ is the negative log-likelihood of the inter-modal similarities with maximum likelihood function defined as formula (5).

$$P\left(S_{ij}^{*} \mid F_{*i}, G_{*j}\right) = \begin{cases} \sigma\left(\theta_{ij}\right), & S_{ij}^{*} = 1 \\ 1 - \sigma\left(\theta_{ij}\right), & S_{ij}^{*} = 0 \end{cases} \tag{5}$$

where $S^* \in \{1, 2\}$, $\sigma\left(\theta_{ij}\right) = \frac{1}{1+e^{-\theta_{ij}}}$ and $\theta_{ij} = 1/2\left(F_{*i}^T * G_{*j}\right)$. Here, $F_{*i} = f(x_i; \theta_x)$ and $G_{*j} = g\left(y_i; \theta_y\right)$ are the feature vectors of the i-th instance of the x-ray image and the j-th instance of the diagnosis report, respectively. It is not hard to find that minimizing the loss function J1 is equivalent to maximizing the likelihood function. Namely, the inner product(similarity) between F_{*i} and G_{*j} is largest when $S_{ij}^1 = 1, S_{ij}^2 = 1$ is the second, and $S_{ij}^2 = 0$ is the smallest. In this way, the distance between samples of the same category is smaller than the distance between samples of similar categories in common space, and the distance between dissimilar categories is the largest. Considering that sample imbalance is a common problem in the medical field, we introduce focal loss optimization to improve the performance of the algorithm. J1 is can be transformed to the following formula (6).

$$\begin{aligned} J1 = -\sum_{i,j=1}^{n} [1 - P(S_{ij}^1|F_{*i}, G_{*j}))^{\gamma} \log(1 - P(S_{ij}^1 \mid F_{*i}, G_{*j} \\ + \mu(1 - P(S_{ij}^2 \mid F_{*i}, G_{*j}))^{\gamma} \log(1 - P(S_{ij}^2 \mid F_{*i}, G_{*j})] \end{aligned} \tag{6}$$

where γ is focusing parameter, $\gamma \geq 0$.

To keep the semantic information in the original space in the hash code, similar to DCMH [4], we also set $B^{(x)} = B^{(y)} = B$, the loss function J2 is defined as formula (7).

$$J2 = \|B - F\|_F^2 + \|B - G\|_F^2 \quad \text{s.t.}\, B \in \{-1, +1\}^{k \times n} \tag{7}$$

Due to F and G can preserve the semantic structure information between S^1 and S^2 by optimizing the J2 term, the similarity between cross-modalities can also be preserved in the hash code we obtained, which exactly matches the goal of cross-modal hashing.

Besides, we also added a balance constraint term J3, which is defined as formula (8).

$$J3 = \|F1\|_F^2 + \|G1\|_F^2 \tag{8}$$

By optimizing the term J3, it is possible to keep the hash code to take "+1" or "−1" on each bit and have the same amount, which to maximize the information on each bit of the hash code. After the above inferences, the final objective function can be formulated as formula (9).

$$\begin{aligned} \min_{\theta_x, \theta_y, B} J &= J1 + \alpha J2 + \beta J3 \\ &= -\sum_{i,j=1}^{n} \left[(1 - P(S_{ij}^1|F_{*i}, G_{*j}))^{\gamma}\ \log\left(1 - P\left(S_{ij}^1 \mid F_{*i}, G_{*j}\right)\right.\right. \\ &\quad + \mu\left(1 - P\left(S_{ij}^2 \mid F_{*i}, G_{*j}\right)\right)^{\gamma} \log\left(1 - P\left(S_{ij}^2 \mid F_{*i}, G_{*j}\right)\right] \\ &\quad + \alpha\left(\|B - F\|_F^2 + \|B - G\|_F^2\right) + \beta\left(\|F1\|_F^2 + \|G1\|_F^2\right) \\ &\quad \text{s.t.}\, B \in \{-1, +1\}^{k \times n} \end{aligned} \tag{9}$$

where $F \in R^{k \times n}$ and $G \in R^{k \times n}$ denote the feature vector matrix for the x-ray images and diagnosis reports, respectively, μ, α and β are hyper-parameters.

4 Optimization Algorithm

We use a strategy of alternating learning to optimize parameters θ_x, θ_y and B. When optimizing one parameter, the other two parameters are fixed. The detailed optimization process derivation will be introduced in detail in this subsection.

4.1 Optimizing $\boldsymbol{\theta_x}, \boldsymbol{\theta_y}$

In the process of learning the feature extraction network of image modalities parameters θ_x, θ_y and B are fixed, which means these two parameters are regarded as constants. For each iteration, a batch size of data is selected from the training data set and input to the network for training. The stochastic gradient descent (SGD) optimization algorithm for each selected sample $\mathrm{x_i}$ is used to learn θ_x. θ_y and θ_x are optimized in the same way.

4.2 Optimizing $\boldsymbol{B}$

In the learning process of the hash code, the parameters θ_x and θ_y be fixed. At this time, formula (9) is equivalent to formula (10).

$$\max_B \operatorname{tr}\left(B^T(\alpha(F+G))\right) = \operatorname{tr}\left(B^T V\right) = \sum_{i,j} B_{ij} V_{ij} \text{ s.t. } B \in \{-1, +1\}^{k \times n} \tag{10}$$

where $V = \alpha(F + G)$. Therefore, B_{ij} should have the same sign with V_{ij} as shown in formula (11).

$$B = \operatorname{sign}(V) = \operatorname{sign}(\alpha(F + G)) \tag{11}$$

During the training process, three parameters, θ_x, θ_y and B, are updated alternately until the specified number of epochs is finished. Then the trained network model has the property of mapping the original input data to the hamming space and maintaining the original semantics.

4.3 Out-of-Sample Extension

After completing the above training, we can generate a corresponding hash code through the trained deep neural network for any modality data of image or text not in the training set. For example, given a query x-ray image x_q, the hash code b_q^x can be obtained using the forward propagation algorithm as formula (12).

$$b_q^x = \operatorname{sign}\left(f\left(x_q; \theta_x\right)\right) \tag{12}$$

Similarly, given a query diagnosis report y_q, the hash code b_q^y can be obtained using the forward propagation algorithm as formula (13).

$$b_q^y = \operatorname{sign}\left(g\left(y_q; \theta_y\right)\right) \tag{13}$$

5 Experiments and Results

5.1 Dataset

MIMIC-CXR Dataset: The MIMIC-CXR dataset is a large publicly available chest X-ray database of 227,835 imaging studies at Beth Israel Deaconess Medical Center (BIDMC) in Boston, Massachusetts. The dataset has a total of 377,110 chest x-ray images in the Digital Imaging and Communications in Medicine (DICOM) format. Each image has its corresponding diagnosis report, which summarizes image-specific annotations by the diagnosis clinician and each image with 14 labels. We adopted 9172 image-text pairs from the database, of which 400 were randomly selected as the query set and others as the set to be retrieved. For the training stage, 6000 pairs from the retrieval set are used as the training set.

5.2 Evaluation Protocol

For cross-modal hashing retrieval, we adopt Mean Average Precision (MAP) and hash lookup to evaluate our method and other baselines. MAP is a combination of precision and recall into a single comprehensive metric. Hash lookup aims to return retrieval data points in a radius of a certain Hamming distance to the given query point. In addition, we also plot precision-recall curves to compare the performance of algorithms.

Table 3. The MAP results for our method and other methods on MIMIC-CXR datasets and best accuracy are shown in boldface.

Task	T->I				I->T			
Bit	16	32	64	128	16	32	64	128
CCA	0.3433	0.3238	0.3453	0.3512	0.3432	0.3207	0.3625	0.3658
CMFH	0.3612	0.3534	0.3602	0.3645	0.3267	0.3324	0.3448	0.3674
SCM	0.3453	0.3676	0.3564	0.3667	0.3655	0.3765	0.3787	0.3763
DCMH	0.6872	0.6710	0.6693	0.6725	0.6879	0.7792	0.7221	0.6980
DSCMR	0.6959	0.6893	0.6828	0.6823	0.7850	0.7863	0.7668	0.7644
SSAH	0.6877	0.6889	0.6754	0.6765	0.7278	0.7723	0.7325	0.7344
SECMR	**0.8165**	**0.8186**	**0.8223**	**0.8561**	**0.8120**	**0.8416**	**0.8620**	**0.8684**

5.3 Implementation Details and Compared Method

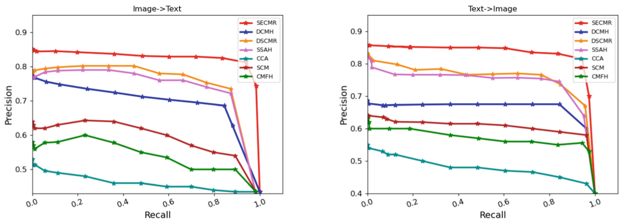
Our proposed method adopts the deep learning open source framework Pytorch, and all experiments are trained on the NVIDIA Tesla P40 GPU server. For hyper-parameters μ, α, and β, we find that the best performance is obtained when $\mu = \alpha = 1, \beta = 0.1$. Hence, we set $\mu = \alpha = 1$ and $\beta = 0.1$ for the proposed method. All other parameters in deep networks are initialized randomly. The inputs of the image modality network are raw x-ray medical images. The secondary labels are utilized as the ground truth. While for diagnosis diagnostic reports, we use a pre-trained BioSent2vec [16] model to map text to a 700-dimensional vector, which serves as the input to the text modality network. We set the mini-batch to 128 and the number of iterations to 100. All experiments were carried out five times, and the final average was taken as the experimental results for analysis.

Compared Method. We compared our method with several state-of-the-art cross-modal hashing methods, including CCA [10], CMFH [15], SCM [14], DCMH [4], SSAH [5] and DSCMR [17]. Among them, CCA and CMFH are classic shallow-structure-based methods, which do not involve deep learning. DCMH, SSAH and DSCMR are classic methods in deep cross-modal hash retrieval, which are recognized as effective methods published at the IEEE Conference on Computer Vision and Pattern Recognition(CVPR) conference in 2017, 2018 and 2019, respectively.

5.4 Experimental Results

Two tasks are performed in this experiment, using text modality as a query to retrieve image modality(T->I) and retrieving text modality data with image modality(I->T), respectively. The MAP results for our method and other methods on MIMIC-CXR datasets are reported in Table 3. It is not difficult to find that our method outperforms the other two methods in any case, average performance increases by 14% and 7% in two retrieval tasks, T->I and I->T, in terms of MAP, respectively. It proves the effectiveness of our designed loss function, our method SECMR retrieves homogeneous case samples and similar categories of samples can be retrieved by semantic extension.

Precision-Recall Curves. Precision-Recall curves with the hash code length as 128 bits on the MIMIC-CXR dataset, as shown in Fig. 2. We can see that our method SECMR consistently achieves the best performance compared to its counterparts. These evaluation results are consistent with the mean average precision.

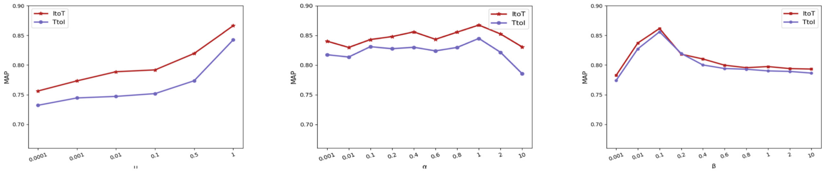


(a) (b)

Fig. 2. Precision-Recall curves on MIMIC-CXR dataset. The hash code length is 128 bits.

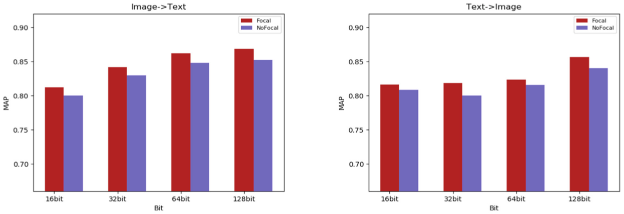
5.5 Parameter Analysis

In this section, we explore the influence of hyper-parameters μ, α, and β on our proposed method. Figure 3 shows the MAP results on the MIMIC-CXR dataset with different values of μ, α, and β, where the code length is 128 bits.



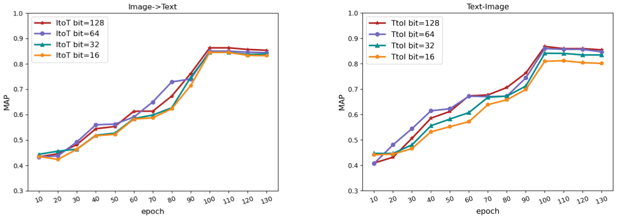
(a) (b) (c)

Fig. 3. sensitivity analysis of the hyper-parameters



(a) (b)

Fig. 4. Effect of Focal loss



(a) (b)

Fig. 5. Training efficiency in terms of MIMIC-CXR datasets

5.6 Effect of Focal Loss

There are some common diseases in the medical field, and some diseases are rare. Under these circumstances, it is not easy to obtain a large-scale and well-distributed dataset to train the neural network, so this paper introduces focal loss to solve this problem. As shown in Fig. 4, We made an experimental comparison of whether to use the focal loss function or not, and the experiments showed that average performance increased by 3% and 1.2% in two retrieval tasks, T->I and I->T in terms of MAP when the focal loss is applied.

5.7 Convergence

Figure 5 shows the trend of two retrieval tasks' MAP scores with the training epochs of our method on MIMIC-CXR datasets. From these figures, we can see that the MAP scores reach their peak within about 100 epochs. In general, our model can converge after training fewer epochs, which illustrates the efficiency of our approach.

6 Conclusion

In this paper, we propose a new cross-modal method (SECMR) to learn a common representation space of medical x-ray images and diagnosis reports. Through semantic expansion, the similarity information between samples in the same category, similar categories, and dissimilar categories is comprehensively considered to ensure that the learned common representation hash code has the multi-level semantic relationship in the original space. The comparable experiments indicate the superiority in retrieval performance of our algorithm on MIMIC-CXR datasets.

Acknowledgement. This research work is supported by the Sci-Tech Innovation 2030 "- New Generation Artificial Intelligence" Major Project (2018AAA0102100); the Natural Science Foundation of Liaoning Province (2021-MS-261).

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2075–2082 (2014)
3. Irie, G., Arai, H., Taniguchi, Y.: Alternating co-quantization for cross-modal hashing. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1886–1894 (2015)
4. Jiang, Q.Y., Li, W.J.: Deep cross-modal hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3232–3240 (2017)
5. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4242–4251 (2018)
6. Lin, Z., Ding, G., Hu, M., Wang, J.: Semantics-preserving hashing for cross-view retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3864–3872 (2015)
7. Liu, H., Feng, Y., Zhou, M., Qiang, B.: Semantic ranking structure preserving for cross-modal retrieval. Appl. Intell. **51**, 1802–1812 (2021)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
9. Pagliardini, M., Gupta, P., Jaggi, M.: Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint arXiv:1703.02507 (2017)
10. Rasiwasia, N., et al.: A new approach to cross-modal multimedia retrieval. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 251–260 (2010)
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)

12. Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 785–796 (2013)
13. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
14. Wang, X., Hu, P., Zhen, L., Peng, D.: DRSL: deep relational similarity learning for cross-modal retrieval. Inf. Sci. **546**, 298–311 (2021)
15. Zhang, D., Li, W.J.: Large-scale supervised multimodal hashing with semantic correlation maximization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
16. Zhang, Y., Chen, Q., Yang, Z., Lin, H., Lu, Z.: BioWordVec, improving biomedical word embeddings with subword information and mesh. Sci. Data **6**(1), 52 (2019)
17. Zhang, Y., Ou, W., Shi, Y., Deng, J., You, X., Wang, A.: Deep medical cross-modal attention hashing. World Wide Web **25**(4), 1519–1536 (2022)
18. Zhen, L., Hu, P., Wang, X., Peng, D.: Deep supervised cross-modal retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10394–10403 (2019)
19. Zhou, J., Ding, G., Guo, Y.: Latent semantic sparse hashing for cross-modal similarity search. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 415–424 (2014)
20. Zhu, L., Tian, G., Wang, B., Wang, W., Zhang, D., Li, C.: Multi-attention based semantic deep hashing for cross-modal retrieval. Appl. Intell. **51**(8), 5927–5939 (2021). https://doi.org/10.1007/s10489-020-02137-w

ECOD: A Multi-modal Dataset for Intelligent Adjudication of E-Commerce Order Disputes

Liyi Chen[1], Shuaipeng Liu[2](✉), Hailei Yan[2], Jie Liu[1](✉), Lijie Wen[3], and Guanglu Wan[2]

[1] College of Artificial Intelligence, Nankai University, Tianjin, China
liyichen@mail.nankai.edu.cn, jliu@nankai.edu.cn
[2] Meituan, Beijing, China
{liushuaipeng,yanhailei,wanguanglu}@meituan.com
[3] School of Software, Tsinghua University, Beijing, China
wenlj@tsinghua.edu.cn

Abstract. With the widespread popularity of e-commerce websites, online order reviews and scores help customers to choose quality products and stores. At the same time, the order dispute problem has gradually attracted attention. Specifically, bad reviews with low scores can have negative impacts on the store, where some unfair and biased reviews mislead other consumers. In order to maintain their reputation, stores usually submit responses against bad reviews. In this paper, we creatively define an intelligent adjudication task of e-commerce order disputes, which aims to judge disputes fairly based on customer reviews and store responses. Moreover, we construct a multi-modal dataset about E-Commerce Order Disputes (ECOD). It contains 6,366 pairs of multi-modal reviews and responses. And each dispute has an adjudication label annotated by business experts. We evaluate the ECOD dataset with baseline models, and analyze the difficulties and challenges in detail. We believe that the proposed dataset will not only facilitate future research in the field of dispute adjudication and multi-modal understanding, but also advance intelligent management for e-commerce websites.

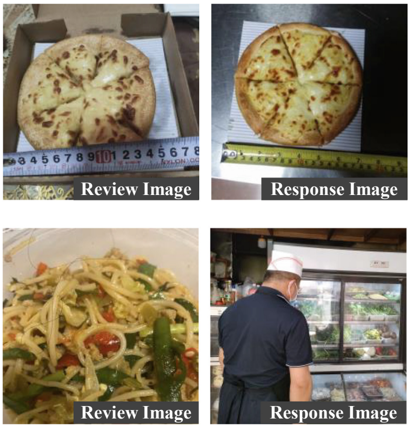
Keywords: Multi-modal dataset · Data classification · Intelligent dispute adjudication

1 Introduction

With the ever-increasing popularity of e-commerce websites, order reviews and scores provide real and detailed information about products, and further influence the purchase decisions of customers [2,18,25]. Good reviews can not only attract more customers but also improve the score of the store, so as to obtain

L. Chen—Work done while Liyi Chen was an intern at Meituan.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 456–468, 2023.
https://doi.org/10.1007/978-3-031-44693-1_36



Example 1

Customer Review: 7寸难道不是23cm吗？这披萨只有5寸。Isn‘t 7 cun (a unit of length) 23 centimeters? This pizza looks only 5 cun.

Store Response: 客户点的是7英寸比萨，拿尺量说没有7寸。产品尺寸是英寸，不是寸。 The customer bought a 7-inch (17.8 cm) pizza, but measured it and found less than 7 cun (23.3 cm). The product size is inches, not cun.

Dispute Adjudication: Support Store

Example 2

Customer Review: 太不卫生了！看到这个东西！简直要吐了！ It‘s too unhygienic! See this thing! I’m throwing up!

Store Response: 本店是男厨师制作炒饭，米线等。我也想不通那来的头发。 In our store, a male chef makes Fried Rice, rice noodles, etc. I can't figure out where the hair came from.

Dispute Adjudication: Support Customer

Fig. 1. Examples for the e-commerce order dispute adjudication task (Chinese is translated into English). The "**Review Image**" and "**Response Image**" are submitted by the customer and store respectively. "**Dispute Adjudication**" refers to the adjudication result.

a higher position in the recommendation list. Conversely, bad reviews can filter defective products and reduce the risk to other consumers. However, many unfair and biased reviews are harmful to the e-commerce environment. For example, some customers give bad reviews to vent their emotions since their unreasonable demands have been rejected. Even there exist malicious competition phenomena. To maintain their reputation, stores can complain and respond directly to these reviews, which expects to eliminate the impact of bad reviews and low scores. With the increasing number of users, order dispute becomes a common problem in the field of e-commerce.

For the large number of order disputes generated daily, the e-commerce website requires a lot of manpower and time to verify them. Moreover, some stores complain all bad reviews indiscriminately to obtain more benefits, which further increases the difficulty and burden of dispute adjudication. As shown in Fig. 1, we present two examples of e-commerce order disputes. For example 1, the customer gives a bad review because the pizza size is not enough. In the response, the store provides a reasonable explanation for the size. Thus, dispute adjudication is "Support Store". In example 2, the response does not clearly confirm that the hair is not related to the store, so dispute adjudication is "Support Customer".

In this paper, we explore the problem of order disputes and define an e-commerce order dispute adjudication task. Specifically, we aim to judge disputes fairly based on customer reviews and store responses. Compared with the existing online review research, such as review sentiment analysis [20,23] and fake review detection [14,16], the research scope of e-commerce order dispute adjudication has two main differences. On the one hand, customers can only submit reviews on their own orders on e-commerce websites, which reduces the phenomenon of fake reviews to a certain extent. On the other hand, order disputes include not only customer review information, but also store response information. Therefore, e-commerce order dispute adjudication is a new research field.

To advance the dispute adjudication task, we construct a multi-modal E-Commerce Order Dispute (ECOD) dataset. First, we define the scope of the dataset and collect more than 15,000 pairs of review-response cases from a popular Chinese e-commerce website. Next, we conduct preliminary data filtering based on specific rules. In addition, we invite business experts to check and annotate each dispute with an adjudication label. Finally, the ECOD dataset has 6,366 order disputes with 14,308 images. And each pair of review-response is multi-modal including text and image. We evaluate the proposed ECOD dataset using the baseline methods. The experimental results show that the proposed dataset is challenging and creates new opportunities for research.

In summary, the contributions of this paper lie in three folds:

– We formally define an intelligent adjudication task of e-commerce order disputes, which aims to fairly judge disputes based on customer reviews and store responses.
– We construct the ECOD dataset[1], involving multi-modal e-commerce order disputes, i.e., review-response pairs. Each dispute has an adjudication label annotated by business experts.
– We conduct experiments on the ECOD dataset using baseline methods, and further analyze the role of different modalities and the challenge of the ECOD dataset. We hope this can advance the research in related fields.

2 Related Work

2.1 Review Classification

In the field of e-commerce, the opinions of reviews usually guide product design, marketing, and other decisions [11]. Most of the previous research about online reviews usually focuses on two tasks: review sentiment analysis and fake review detection.

Review sentiment analysis [1,6] aims to understand the underlying sentiment of text and images, such as positive and negative. It is widely used in many fields, such as book reviews [22], movie reviews [10], product reviews [11,12], etc. Analyzing the sentiment of online reviews can help to summarize user preferences, suggest new products that might enjoy, and discover other potential customers [21]. In recommender systems, the scores of online reviews can also recommend high-quality products to users. For example, McAuley and Leskovec [11] analyze user reviews of products, including multiple top-level categories of products, such as books and movies, to discover latent product and user dimensions.

[1] https://github.com/LiyiLily/ECOD.

In addition, McAuley and Leskovec [12] explore user reviews of different foods to understand their tastes and their level of experience.

Since the significant impact of online reviews on consumption decisions, it incentivizes many fake reviews to promote or discredit target products. Following web spam and email spam tasks, Jindal and Liu [7] propose a fake review detection task that aims to identify opinion spam in reviews. Inspired by this, more studies [13,17,19] explore the opinion spam issue in online review websites. Many researches [15,19] identify suspicious users and reviews based on the text features of reviews and the behavioral features of users. We noticed that the above tasks generally focus on online review systems, such as Yelp[2] and Dianping, where every user can submit a review for any store without an order form.

On e-commerce websites, customers are only allowed to submit reviews for their orders. This alleviates the fake review problem to a large extent but faces a new dispute problem between customers and stores.

Table 1. Comparison with existing online review datasets. "CR" and "SR" denote "Customer Review" and "Store Response". Annotation means how the data is annotated, such as Amazon Mechanical Turk (AMT), filtering algorithm (FA), rule-based method (RB), and manual annotation (MA).

Dataset	Domain	Language	Order	CR	SR	Modality	Annotation
TripAdvisor [17]	hotel	English	✗	✓	✗	text	AMT
YelpChi [15]	restaurant/hotel	English	✗	✓	✗	text	FA
YelpNYC [19]	restaurant	English	✗	✓	✗	text	FA
YelpZip [19]	restaurant	English	✗	✓	✗	text	FA
Amazon [7]	e-commerce	English	✓	✓	✗	text	RB
Dianping [9]	restaurant	Chinese	✗	✓	✗	text	FA
ECOD	e-commerce	Chinese	✓	✓	✓	text+image	MA

2.2 Relevant Datasets

Based on existing research on online reviews, relevant datasets are often collected from popular online review websites and e-commerce websites. Jindal and Liu [7] analyze three types of spam reviews on Amazon[3], including untruthful opinions, reviews on brands only, and non-reviews. Ott et al. [17] mine truthful reviews about 20 most popular hotels on TripAdvisor[4] and generate deceptive positive reviews using Amazon Mechanical Turk (AMT). Subsequently, Mukherjee et al. [15] propose that there is a precise psycholinguistic difference between AMT reviews and truthful reviews. Some studies construct datasets from truthful reviews, such as YelpChi [15], YelpNYC [19], YelpZip [19], etc. They regard

[2] https://www.yelp.com.

[3] http://www.amazon.com.

[4] https://www.tripadvisor.com.

the filtered reviews in online review websites as fake, and the unfiltered reviews as non-fake. However, most relevant datasets pay more attention to online reviews and user behaviors, but lack the response of stores. Therefore, existing datasets containing only customer reviews are inapplicable for the order dispute adjudication task.

In Table 1, we summarize existing relevant datasets. The proposed ECOD dataset is: (1) more comprehensive - the dataset contains not only customer reviews but also store responses, providing more comprehensive descriptions of order disputes. (2) more reliable - the dataset is collected from a popular e-commerce website where each review-response pair is rely on a real order form, and the adjudication label is annotated by business experts. (3) multi-modal - compared with existing review datasets, the ECOD dataset contains both text and image modalities.

3 Dispute Adjudication Task

3.1 Order Dispute Description

On e-commerce websites, customers can only submit reviews for their orders, which enhances the reliability of reviews to a certain extent. Specifically, an online review consists of a combined score and a detailed review that supports both text and image modalities. In general, a good review usually means to recommend the product, and a bad review to not recommend it. Objective and truthful reviews can help other customers choose high-quality products and guide consumption decisions.

However, some disputable bad reviews with low scores may mislead consumers. For example, (1) a review is positive but has a low score, (2) a customer submits bad reviews to vent dissatisfaction since their unreasonable demands are rejected by the store, (3) a review is irrelevant to the target order and target store, (4) etc. To maintain their reputation, stores usually submit responses directly and make complaints on e-commerce websites against disputable bad reviews. Since the huge number of customers and stores on e-commerce websites, many disputes arise every day. Handling disputes manually not only consumes a lot of manpower but also limits the speed of dispute resolution. Furthermore, some stores make complaints for all bad reviews indiscriminately, increasing the level of challenge to the order dispute adjudication task.

Compared with the existing review classification research, the proposed order dispute adjudication task has three main characteristics. First, the dispute adjudication task is multi-modal, where customers and stores can upload images to enhance the authenticity of the text. Second, judging a dispute considers not only the customer review but also the store response. Both stores and customers can express their opinions, instead of one-way reviews from customers. In the store responses, stores need to explain and prove the unreasonableness of bad reviews. Third, for each customer and store, the order dispute is a rare event. It means the user behavior feature is not distinguishable. It is difficult to judge disputes based on the identity of customers and stores, so the content of reviews and responses is critical in this task.

3.2 Task Definition

In this paper, we formally define an intelligent adjudication task of e-commerce order disputes, which aims to judge disputes between customers and stores fairly. Specifically, we predict the adjudication label of dispute using the review and response information. The order dispute adjudication task can be formulated as:

$$\{X_T^C, X_I^C, X_T^S, X_I^S\} \xrightarrow{classify} Y, \tag{1}$$

where X^C refers to the customer review, and X^S refers to the store response. T and I represent text and image modal input, respectively. And Y is the adjudication result.

Table 2. The description of each dispute type on ECOD dataset.

Dispute Type	Description
Nonsense review	The review does not explain the reason for the low score
Unreasonable demands	The demand of the customer exceeds the service scope of the store
Menace	Customers threaten stores with low scores, expecting to profit from it
Intratype competition	The customer is suspected of malicious competition resulting in a low score
Irrelevant reviews	The content of the review is inconsistent with the product in the order
Mismarking	The review is positive, but the customer gives a low score
No special requests	The customer does not leave any special requests in the order
Pro-environment order	The customer chooses the "pro-environment order", resulting in no cutlery
Unreceived order	The customer gave a low score before receiving the order
Delivery delay	The store completes the order on time, but the delivery man delivers it late
Poor delivery service	The delivery man has a poor service attitude
Other delivery issues	Other issues of the delivery man resulted in a low score
Abusive reviews	The review includes abusive text or advertising
Other	Other reviews with low scores

4 Dataset

4.1 Dataset Overview

In the paper, we construct ECOD, a multi-modal e-commerce order dispute dataset, from a Chinese e-commerce website. The dataset has 6,366 dispute cases with 14,308 images. Each dispute case includes both customer review and store response, which associates with a real disputable order. Specifically, the reviews are submitted by customers describing their dissatisfaction with the disputable order or explaining why they give a low score. And the responses are submitted by stores, which aim to emphasize the low score is unreasonable. Moreover, each review-response pair is multi-modal involving text and image. The images can enhance the authenticity of the relevant review or response, and further increase the possibility of obtaining a favorable adjudication result. Each review-response pair has an adjudication label annotated by business experts. For the ECOD dataset, the adjudication results have the following three labels:

– **Support Customer (SupCus)**: The bad review and low score are objective and reasonable. The store is responsible for the dispute.
– **Neutrality (Neutral)**: The dispute caused by force majeure. Neither the customer nor the store is responsible.
– **Support Store (SupSto)**: The bad review and low score are biased and unreasonable. The customer is responsible for the dispute.

4.2 Dataset Construction

We collect and annotate the ECOD dataset with the following three stages: (1) data collection, (2) preliminary data filtering, and (3) professional data annotation.

Data Collection. We collect disputable order information from a popular Chinese e-commerce website, which has a critical mass of Chinese customers and stores. We capture the disputable orders which have a low score and contain the store response information. Notably, some disputable orders are beyond the scope of the proposed dispute adjudication task, which is difficult to judge disputes only using review-response pairs. For example, some suspicious customers give low scores for a store continuously. Therefore, we initially identify the scope of disputable orders in data collection phase to ensure that most of the collected data are of research significance. As shown in Table 2, we select 14 task-related dispute types.

For each disputable order, we extract the relevant review-response pair, including text and image data. In the end, we collect more than 15,000 pairs of order disputes, where each dispute contains both review images and response images.

Preliminary Data Filtering. We filter the collected dispute cases to control the data quality.

We convert the image format to JPG uniformly and remove data containing corrupted images. Since some reviews and responses include rich-text images, such as screenshots of order information, which do not belong to the research scope of this paper and is easy to make privacy disclosure, we filter the dispute data that has rich-text images. In order to distinguish between rich-text images and natural images (such as product photos), we perform binary classification on the collected images. Specifically, we manually assign 1,000 randomly selected

Table 3. Statistics of the ECOD dataset. The average text length is calculated by the number of Chinese words.

	SupCus	Neutral	SupSto	Total
Train set	2,809	1,396	889	5,094
Test set	702	348	222	1,272
Total	3,511	1,744	1,111	6,366
Avg # images per dispute	2.24	2.25	2.27	2.25
# of unique customers	3,507	1,743	1,111	6,359
# disputes per customer	1.00	1.00	1.00	1.00
Avg # words per customer review text	31.46	25.33	24.40	28.55
# of unique stores	3,440	1,675	1,100	6,129
# disputes per store	1.02	1.04	1.01	1.04
Avg # words per store response	41.70	33.55	48.88	40.72

images to the rich-text image and the natural image, where the training set and the validation set contain 800 and 200 images respectively. Based on the ResNet-50 [5] model, we fine-tune the parameters of the last classification layer and finally achieve 98.5% accuracy on the validation set. Using the fine-tuned model, rich-text images can be filtered preliminarily. In the next data annotation stage, we also filter rich-text images manually to make sure privacy security.

Moreover, we delete duplicate data based on the text information of reviews and responses. Furthermore, we discard some dispute data that are too easily predicted. For example, in actual dispute adjudication, if the text content is empty that cannot explain the reason for the low score, the adjudication result will often tend to favor the store. Conversely, if the response is ambiguous and cannot refute the bad review and low score, the adjudication result will be in favor of the customer. Therefore, we filter out some dispute cases which have uncertain meanings in review or response texts, e.g., the review or response text is empty, the text length is less than 5, the text has many garbled characters, etc.

Professional Data Annotation. We invited 20 business experts to manually filter data and annotate the ECOD dataset. All the business experts are Chinese native speakers and are experienced in handling e-commerce order disputes between customers and stores. For each dispute case, 5 business experts give an adjudication label based on customer reviews and store responses. When at least 4 experts give the same annotation results, the disputed case can be retained in the dataset. At this stage, business experts further filter the data beyond the scope of the study and the data with rich-text images. Moreover, private data in text and images is manually anonymized, including names, phone numbers, and other risk information. Finally, the ECOD dataset has 6,366 dispute samples, and the labeling consistency rate of experts is about 92%.

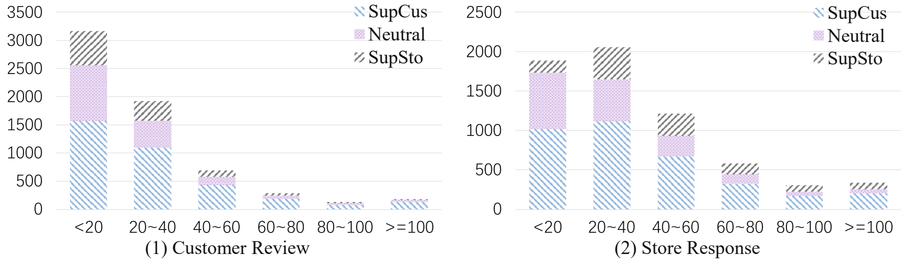


Fig. 2. Length distributions of customer review text and store response text.

4.3 Dataset Statistics

According to Table 3, the dataset contains a total of 6,366 disputes involving 6,359 customers and 6,129 stores. The "Support Customer", "Neutrality", and "Support Store" classes involve 3,511, 1,744, and 1,111 dispute cases, respectively. Each dispute case has an average of 2.25 images, including at least one review image and at least one response image. It is worth noting that the average number of disputes for each customer and store is less than 1.1, which means the proposed dispute adjudication task cannot be simply transformed into detecting suspicious users. It is difficult to predict the adjudication label based on user behavior, historical orders, and historical disputes. In addition, we observe that the longer text is more conducive to obtaining support in the adjudication result. For example, the disputes in the label "Support Customer" have longer review texts (avg. 31.46 words), and the disputes in the label "Support Store" have longer response texts (avg. 48.88 words). Furthermore, we do a survey of length distributions of customer review text and store response text in Fig. 2. As the length of customer review text increases, the proportion of adjustment results that are "Support Customer" or "Neutrality" gradually increases. And when the length of the store response text is less than 20, the adjustment result is rarely "Support Store".

5 Experiment

We adopt baseline models to comprehensively evaluate the proposed ECOD dataset. We first perform experiments based on text modality features and then combine visual modality features. Furthermore, we analyze the difficulties and challenges of the ECOD dataset based on the case study.

5.1 Evaluation Metrics

The e-commerce order dispute adjudication is a classification task with three adjudication labels, i.e., "Support Customer", "Neutrality", and "Support Store". Following the existing classification tasks [9,17], we employ several comprehensive metrics to perform evaluation, including **Precision**, **Recall**, **F1-score**, and **Accuracy**.

5.2 Baseline Method

For the text modality data, we capture the semantic feature using the Chinese BERT [4], which includes an embedding layer and a stack of Transformer layers. The input text modality data is illustrated as follows,

$$[CLS]\ Review\ Text\ [SEP]\ Response\ Text\ [SEP]. \tag{2}$$

The review text and response text are concatenated together into a single sequence using a special token "[SEP]". Specially, we use segment embedding to indicate whether every token belongs to the review text or response text. In addition, we add a special token "[CLS]" at the beginning of the sequence, which is used to aggregate text representation and predict dispute adjudication results. The dimension of the final text representation is d.

For the visual modality data, we use ResNet-50 (ResNet) [5] to extract the image feature, which is a popular Convolutional Neural Network (CNN) and widely used for multiple image classification tasks, such as fine-grained image classification, sentiment classification, etc. We extract image features of review and response input separately. For a dispute instance, we first calculate the features of each review image to dimension d as same as the text feature. And we obtain the average of multiple review images as the final review image representation. Similarly, we calculate the response image representation. Then, we concatenate the review and response image representation, and aggregate all image representations into dimension d using a fully connected layer.

Finally, we use the Softmax activation function to calculate the probability of each label. In multi-modal experiments, we compute the predicted probabilities using text and image representations respectively, and then perform a weighted sum of the predicted probabilities for the two modalities. We adopt a cross entropy loss function in the model.

Table 4. Precision, recall, F1-score, and Accuracy of adjudication result on the ECOD dataset. The boldface indicates the best results. "CT", "ST", "CI", and "SI" refer to the customer review text, store response text, customer review image, and store response image respectively.

Model	Feature	Precision	Recall	F1-score	Accuracy
BERT	CT	0.5784	0.5468	0.5583	0.6281
	ST	0.6105	0.5930	0.5941	0.6486
	CT+ST	0.6223	0.6398	0.6297	0.6690
BERT+ResNet	CT+ST+CI	0.6420	0.6780	0.6539	0.6785
	CT+ST+SI	0.6424	0.6642	0.6513	0.6800
	CT+ST+CI+SI	**0.6511**	**0.6730**	**0.6601**	**0.6918**

5.3 Implementation Details

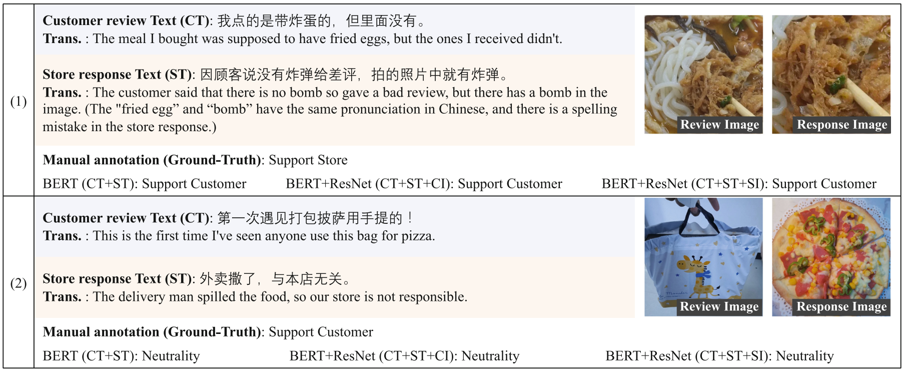
Our code is implemented based on Pytorch and Huggingface Transformers [24]. We use the pre-trained Chinese BERT-base model [4] to initialize the text representation model. The ResNet-50 [5] model is used to extract image representation, which is pre-trained on the ImageNet [3]. We freeze the parameters before the fully connected layer in ResNet-50. All the experiments are conducted using one 32G NVIDIA V100 GPU. By default, we set batch size as 64, learning rate to be 1e-5, dropout rate as 0.1, and a maximum of 35 epochs for training with the early stopping. We use Adam optimizer [8] to minimize the loss function. The maximum character length of both review and response text is set to 128. The maximum number of images is fixed at 2. The size of images is fixed to 224 × 224. The weights of predicted probabilities for text and image are set to 1 and 0.1 based on multiple experiments. The dimension of both text and image representations is 768.

5.4 Experimental Results and Analysis

In Table 4, we show the experimental results on the ECOD dataset using different models and inputs, where “CT”, “ST”, “CI”, and “SI” refer to the customer review text, store response text, customer review image, and store response image respectively. Compared with the BERT model using review text, the BERT model using response text achieves better performance, in which the F1-score increases from 55.83% to 59.41%, and the accuracy increases from 62.81% to 64.86%. Moreover, for the plain text model, using both review and response text can further improve the dispute adjudication result, where the F1-score is raised to 62.97% and the accuracy is raised to 66.90%. Furthermore, by leveraging the multi-modal review and response data, the BERT+ResNet model has an additional performance improvement, where the F1-score increases from 62.97% in the plain text model to 66.01%, and the accuracy achieves 69.18%.

As shown in Fig. 3, we show two common error cases on the ECOD dataset. Since customer reviews and store responses are usually colloquial in the dispute adjudication task, some omissions and spelling mistakes are common in the text. In case (1), because of complex text semantics, models incorrectly predict it as the label “Support Customer”. In case (2), the customer considers that using the bag in the image to pack the pizza will make the pizza tilt and deform, so the store should not use such packaging. However, due to the omitted expression of the review text and the seemingly normal review image, the sample is predicted to be a wrong result.

Finally, we summarize three main challenges of the ECOD dataset: (1) Multiple input sources, ECOD includes customer review input and store response input; (2) Multi-modal data, both customer review and store response are multi-modal; (3) Data imbalance, the adjudication label is imbalanced.



(1)

Customer review Text (CT): 我点的是带炸蛋的，但里面没有。
Trans. : The meal I bought was supposed to have fried eggs, but the ones I received didn't.

Store response Text (ST): 因顾客说没有炸弹给差评，拍的照片中就有炸弹。
Trans. : The customer said that there is no bomb so gave a bad review, but there has a bomb in the image. (The "fried egg" and “bomb” have the same pronunciation in Chinese, and there is a spelling mistake in the store response.)

Manual annotation (Ground-Truth): Support Store

BERT (CT+ST): Support Customer BERT+ResNet (CT+ST+CI): Support Customer BERT+ResNet (CT+ST+SI): Support Customer

(2)

Customer review Text (CT): 第一次遇见打包披萨用手提的！
Trans. : This is the first time I've seen anyone use this bag for pizza.

Store response Text (ST): 外卖撒了，与本店无关。
Trans. : The delivery man spilled the food, so our store is not responsible.

Manual annotation (Ground-Truth): Support Customer

BERT (CT+ST): Neutrality BERT+ResNet (CT+ST+CI): Neutrality BERT+ResNet (CT+ST+SI): Neutrality

Fig. 3. Common error cases of BERT and BERT+ResNet model on the ECOD dataset.

6 Conclusion

In this paper, we define an intelligent adjudication task of e-commerce order disputes, which aims to judge disputes between customers and stores based on the review-response pairs. To facilitate the research, we propose ECOD, a multi-modal e-commerce order dispute dataset with manual annotation, which has 6,366 pairs of multi-modal dispute instances and 14,308 images. We conduct baseline experiments on the ECOD dataset to evaluate the role of different inputs. Furthermore, we analyze the experimental results and summarize the challenge of the dataset. We expect this work will promote the intelligent dispute adjudication task and advance future research on multi-modal understanding.

Acknowledgements. This research is supported by the National Natural Science Foundation of China under grant No. 61976119.

References

1. Chen, C., Teng, Z., Wang, Z., Zhang, Y.: Discrete opinion tree induction for aspect-based sentiment analysis. In: ACL, pp. 2051–2064 (2022)
2. Chen, Y., Xie, J.: Online consumer review: word-of-mouth as a new element of marketing communication mix. Manage. Sci. **54**(3), 477–491 (2008)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: ACL, pp. 4171–4186 (2019)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
6. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Exploiting document knowledge for aspect-level sentiment classification. In: ACL, pp. 579–585 (2018)
7. Jindal, N., Liu, B.: Opinion spam and analysis. In: WSDM, pp. 219–230 (2008)

8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Li, H., Chen, Z., Liu, B., Wei, X., Shao, J.: Spotting fake reviews via collective positive-unlabeled learning. In: ICDM, pp. 899–904 (2014)
10. Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: ACL, pp. 142–150 (2011)
11. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: ACM Conference on Recommender Systems, pp. 165–172 (2013)
12. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: WWW, pp. 897–908 (2013)
13. Mohawesh, R., et al.: Fake reviews detection: a survey. IEEE Access **9**, 65771–65802 (2021)
14. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: WWW, pp. 191–200 (2012)
15. Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.: What yelp fake review filter might be doing? In: AAAI, vol. 7 (2013)
16. Nilizadeh, S., Aghakhani, H., Gustafson, E., Kruegel, C., Vigna, G.: Lightning talk - think outside the dataset: finding fraudulent reviews using cross-dataset analysis. In: WWW, pp. 1288–1289 (2019)
17. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: ACL, pp. 309–319 (2011)
18. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowl. Based Syst. **89**, 14–46 (2015)
19. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: SIGKDD, pp. 985–994 (2015)
20. Serrano-Guerrero, J., Olivas, J.A., Romero, F.P., Herrera-Viedma, E.: Sentiment analysis: a review and comparative analysis of web services. Inf. Sci. **311**, 18–38 (2015)
21. Sharma, A., Cosley, D.: Do social explanations work? Studying and modeling the effects of social explanations in recommender systems. In: WWW, pp. 1133–1144 (2013)
22. Srujan, K., Nikhil, S., Raghav Rao, H., Karthik, K., Harish, B., Keerthi Kumar, H.: Classification of amazon book reviews based on sentiment analysis. In: INDIA, pp. 401–411 (2018)
23. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Aspect-level sentiment analysis via convolution over dependency tree. In: EMNLP-IJCNLP (2019)
24. Wolf, T., et al.: HuggingFace's transformers: state-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
25. Zhang, Z., Dong, Y., Wu, H., Song, H., Deng, S., Chen, Y.: Metapath and syntax-aware heterogeneous subgraph neural networks for spam review detection. Appl. Soft Comput. **128**, 109438 (2022)

Bounding and Filling: A Fast and Flexible Framework for Image Captioning

Zheng Ma[1], Changxin Wang[1], Bo Huang[1], Zixuan Zhu[2],
and Jianbing Zhang[1](✉)

[1] Nanjing University, Nanjing, China
{maz,cx.wang,191300018}@smail.nju.edu.cn, zjb@nju.edu.cn
[2] University of Glasgow, Glasgow, UK

Abstract. Most image captioning models following an autoregressive manner suffer from significant inference latency. Several models adopted a non-autoregressive manner to speed up the process. However, the vanilla non-autoregressive manner results in subpar performance, since it generates all words simultaneously, which fails to capture the relationships between words in a description. The semi-autoregressive manner employs a partially parallel method to preserve performance, but it sacrifices inference speed. In this paper, we introduce a fast and flexible framework for image captioning called **BoFiCap** based on bounding and filling techniques. The BoFiCap model leverages the inherent characteristics of image captioning tasks to pre-define bounding boxes for image regions and their relationships. Subsequently, the BoFiCap model fills corresponding words in each box using two-generation manners. Leveraging the box hints, our filling process allows each word to better perceive other words. Additionally, our model offers flexible image description generation: 1) by employing different generation manners based on speed or performance requirements, 2) producing varied sentences based on user-specified boxes. Experimental evaluations on the MS-COCO benchmark dataset demonstrate that our framework in a non-autoregressive manner achieves the state-of-the-art on task-specific metric CIDEr (125.6) while speeding up 9.22× than the baseline model with an autoregressive manner; in a semi-autoregressive manner, our method reaches 128.4 on CIDEr while a 3.69× speedup.

Keywords: Image Captioning · Non-Autoregressive · Knowledge Distillation

1 Introduction

Image captioning tasks require models to automatically generate a sentence that describes a given image. Thanks to advancements in computer vision [11,22] and natural language processing [13,25], numerous captioning models can accurately describe images [2,5,27]. The application of image captioning extends to various fields, including facilitating language learning in children, aiding visually

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 469–481, 2023.
https://doi.org/10.1007/978-3-031-44693-1_37

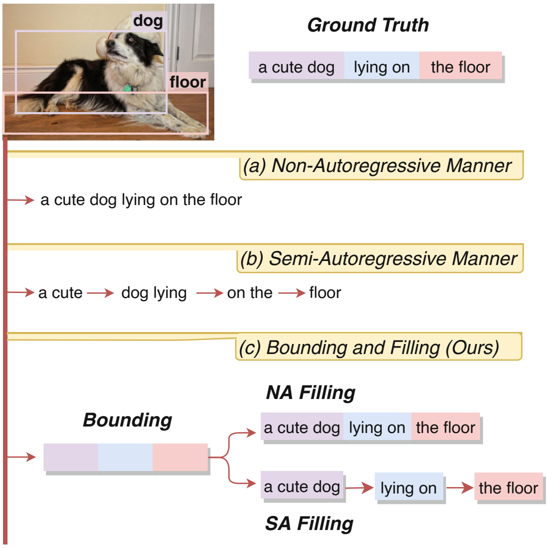


Fig. 1. (Top) exhibits a description that can be split into several boxes, each of which describes a region in the image or a relation between two regions. (Bottom) exhibits the difference between non-autoregressive manner, semi-autoregressive manner, and our BoFiCap. Our model bounds a series of boxes in advance, and then flexibly fills words in these boxes using NA or SA filling manner.

impaired individuals in comprehending their environment, and alerting drivers to potential hazards during autonomous driving.

In a real-time application, inference speed is a crucial factor to consider as models need to provide quick responses. However, many high-performing captioning models utilizing an autoregressive manner [14,18,24] generate descriptions incrementally, word by word, resulting in significant inference latency. Researchers have recognized this challenge and have employed non-autoregressive [6,7,9,10] or semi-autoregressive manner [8,31] to speed up. The non-autoregressive manner generates a description in parallel, as displayed in Fig. 1(a), which greatly improves the inference speed but often leads to performance degradation, repetition, and omissions [8]. These issues arise due to the absence of inter-word dependencies since each word is predicted simultaneously, leading to a limited understanding of its contextual role within the description. The semi-autoregressive manner combines parallel and serial generation, as illustrated in Fig. 1(b), striking a balance between speed and performance. However, it only produces a fixed and semantically meaningless chunk at each step.

It is observed that descriptions typically employ a noun phrase to depict an image region and utilize a conjunctive or verb phrase to articulate the relationship between two regions. As depicted in Fig. 1 (Top), *'a cute dog'*, and *'the floor'* are noun phrases, with each phrase describing a distinct region in the image. *'lying on'* is a verb phrase that articulates the relationship between the two noun phrases. If certain boxes representing regions or the relationship between regions can be predefined, the captioning model can generate descriptions based on the arrangement of these boxes. By doing so, words can establish mutual understanding through their association with specific boxes. Drawing inspiration from this concept, in this paper we propose a novel framework for fast and flexible image captioning via bounding and filling named BoFiCap. To begin with, BoFiCap establishes a series of bounding boxes to facilitate description generation. During the decoding phase, BoFiCap utilizes two filling methods, namely, non-autoregressive (NA) filling and semi-autoregressive (SA) filling, to populate the boxes with the appropriate words. As illustrated in Fig. 1(c), NA filling corresponds to the non-autoregressive manner, simultaneously populating all boxes, while SA filling aligns with the semi-autoregressive manner, progressively populating each box in sequential steps. Regarding the distinction in decoding between the two filling methods, SA filling exhibits the better ability to capture word dependencies compared to NA filling, as the SA method fills boxes incrementally, thus allowing it to leverage boxes already populated with words. Hence, we introduce an imitation strategy wherein NA filling imitates the behavior of SA filling to improve its capability to comprehend other words. Moreover, BoFiCap has the capability to generate diverse descriptions in a flexible manner by choosing various filling methods or providing different arrangements of boxes. The main contributions of this paper are outlined as follows:

- We propose BoFiCap, a fast and flexible framework for image captioning that decouples the generating process into bounding and filling stages. BoFiCap utilizes advanced bounding techniques to define boxes and subsequently fills them using either the NA or SA filling approach. Additionally, our model offers the capability to generate diverse descriptions for the same image in a flexible manner.
- To enhance the performance of BoFiCap, we introduce parameter sharing between the decoders of the NA and SA methods. Furthermore, we propose an imitating strategy that improves the ability of NA filling to capture word dependencies.
- Experimental results demonstrate the effectiveness of our approach. In a non-autoregressive manner, our method achieves state-of-the-art performance while achieving a 9.22× speedup compared to the baseline model. In a semi-autoregressive manner, our method achieves a CIDEr score of 128.4, accompanied by a 3.69× speedup.

2 Preliminary

In this section, we will briefly review three decoding manners in literature, including autoregressive manner, non-autoregressive manner, and semi-autoregressive manner. Given an image I, the image captioning task is requested to generate a description $S = \{w_1, w_2, \ldots, w_T\}$, with T denoting the total length of this description.

Autoregressive Manner. In an autoregressive manner, the output of the t-th step depends on the previous sequence, and the probability of a sequence is the combination of the probabilities of all words. It can use the chain rule to decompose the likelihood of sequences:

$$P(S|I) = \prod_{t=1}^{T} P(w_t|w_{<t}, I), \tag{1}$$

where $w_{<t} = \{w_1, w_2, \ldots, w_{t-1}\}$ represents the generated words before step t.
Non-autoregressive Manner. The non-autoregressive manner is proposed to address the high inference latency problem. It breaks dependency on previously generated words and generates them in parallel, which can be formulated as:

$$P(S|I) = \prod_{t=1}^{T} P(w_t|I). \tag{2}$$

Semi-autoregressive Manner. In a semi-autoregressive manner, a group of words is generated in one step, which is parallel in the group and autoregressive between groups. Assuming S can be divided in $\{g_1, g_2, \ldots, g_N\}$, where N is the number of groups, the semi-autoregressive manner can be formulated as:

$$P(S|I) = \prod_{t=1}^{N} P(g_t|g_{<t}, I), \tag{3}$$

where $g_{<t} = \{g_1, g_2, \ldots, g_{t-1}\}$ is the generated groups before step t.

3 Proposed Method

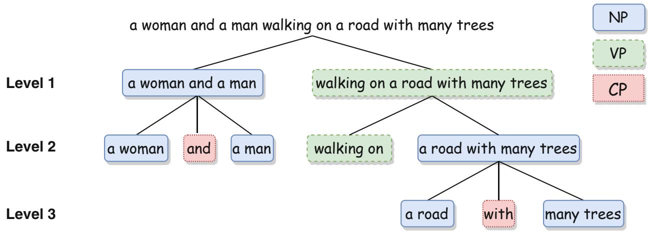


Fig. 2. A description is parsed into a tree structure by a constituency parser.

3.1 Generating Hierarchical Boxes

We use a constituency parser [30] to split sentences in advance. To be specific, a description is parsed into a hierarchical tree structure in which the layers from shallow to deep represent coarse-to-fine bounding information. As shown in Fig. 2, a whole description is first divided into *'a woman and a man'* and *'walking on a road with many trees'*. Then, the two items can be divided into finer components. We define three types of boxes: NP-box, VP-box, and CP-box, corresponding to NP, VP, and CP labels in the constituency parser We name them level-k to distinguish different levels, where $k = -1$ represents all phrases that can no longer be cut.

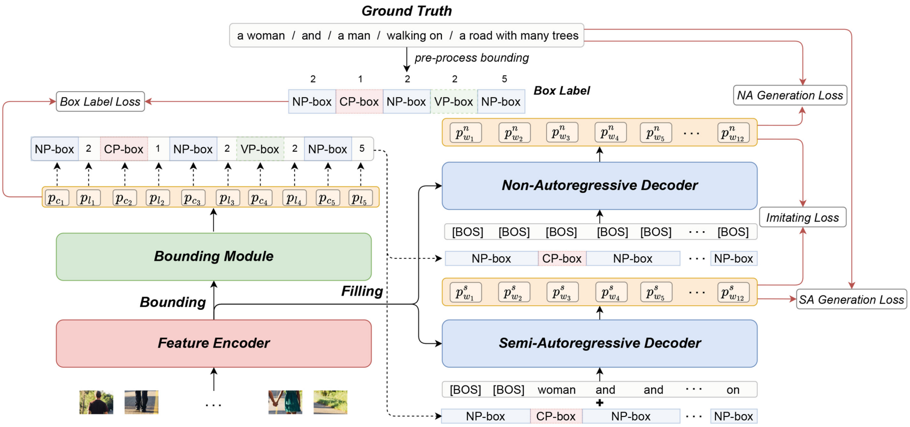

Ground Truth
a woman / and / a man / walking on / a road with many trees
pre-process bounding
2 1 2 2 5
NP-box CP-box NP-box VP-box NP-box
Box Label
Box Label Loss
NA Generation Loss
NP-box 2 CP-box 1 NP-box 2 VP-box 2 NP-box 5
p_{c_1} p_{l_1} p_{c_2} p_{l_2} p_{c_3} p_{l_3} p_{c_4} p_{l_4} p_{c_5} p_{l_5}
$p^n_{w_1}$ $p^n_{w_2}$ $p^n_{w_3}$ $p^n_{w_4}$ $p^n_{w_5}$ ··· $p^n_{w_{12}}$
Non-Autoregressive Decoder
Imitating Loss
[BOS] [BOS] [BOS] [BOS] [BOS] ··· [BOS]
NP-box CP-box NP-box ··· NP-box
Bounding Module
Bounding
Filling
$p^s_{w_1}$ $p^s_{w_2}$ $p^s_{w_3}$ $p^s_{w_4}$ $p^s_{w_5}$ ··· $p^s_{w_{12}}$
SA Generation Loss
Feature Encoder
Semi-Autoregressive Decoder
[BOS] [BOS] woman and and ··· on
+
NP-box CP-box NP-box ··· NP-box
···

Fig. 3. The illustration of our BoFiCap. (Low Left) exhibits the regions are encoded by the feature encoder. (Upper Left) exhibits the process of bounding, it will predict bounding boxes information for filling. (Right) exhibits our BoFiCap for NA and SA filling manners. Noted that the non-autoregressive and semi-autoregressive decoder share parameters.

3.2 BoFiCap Model Architecture

Generally, our BoFiCap model is built on Transformer [25]. As displayed in Fig. 3, it consists of three modules: feature encoder, bounding module, and filling module.

Feature Encoder. The feature encoder aims to encode the image into a visual context. In our framework, we extract regions (denoted as R) from raw images in advance by a faster-RCNN model [2] which uses bottom-up attention and is trained on Visual Genome datasets [17]. The architecture of the feature encoder is as same as the encoder of the vanilla Transformer [25], but we remove the position encoding because regions cannot be arranged linearly.

Bounding Module. To bound a series of boxes, three aspects should be considered in our bounding module: 1) the number of boxes, denoted as N. 2) the number of words in each box, denoted as $L = \{l_1, l_2, \ldots, l_N\}$. 3) the type of each box, denoted as $C = \{c_1, c_2, \ldots, c_N\}$. Further, we can define bounding information as $B = \{b_1, b_2, \ldots, b_N\}$, where $b_i = \{\underbrace{c_i, c_i, \ldots, c_i}_{l_i}\}$, thus enabling B to integrate the above three aspects of information.

Our bounding module generates bounding information autoregressively, which consists of one Transformer decoder layer and two linear classifiers to predict the type of boxes and the number of corresponding words. Specifically, conditioning on the image regions extracted from raw images, our bounding module predicts the bounding information of one box, namely the type of one box and the number of corresponding words, in one step. The probability of bounding information can be factorized as:

$$P(B|R) = \prod_{t=1}^{N} P(b_t|b_{<t}, R) = \prod_{t=1}^{N} P(l_t, c_t|b_{<t}, R). \tag{4}$$

Filling Module. Our filling module includes two decoders with different manners: the non-autoregressive decoder and semi-autoregressive decoder for NA and SA filling respectively, which are as same as the decoder of the vanilla Transformer architecturally. Three pieces of information are input in our filling module, including visual context, bounding information and history words.

In SA filling manner, the SA decoder will fill one box with words in one step. The target description probability can be factorized as:

$$P(S|B, R) = \prod_{t=1}^{N} P(g_t|g_{<t}, b_{\leq t}, R), \tag{5}$$

where g_t represents the filled words in box b_t. Note that the number of words in each box b_i may be different, which means that the generated words at the $t-1$-th step can not be directly regarded as the input at the t-th step. To address this issue, we propose a copy strategy called Position-wise Copy. Intuitively, the two words that are closer in position are likely to be closer in relation. Assuming the output of last step is $g_{t-1} = \{w_1, w_2, \ldots, w_{l_{t-1}}\}$, and current step's output is $g_t = \{w_1^{'}, w_2^{'}, \ldots, w_{l_t}^{'}\}$, each word w_i in g_{t-1} will be copied n_i times by Position-wise Copy:

$$n_i = \begin{cases} \lfloor l_t/l_{t-1} \rfloor, & i \leq l_{t-1} - l_t \% l_{t-1}, \\ \lfloor l_t/l_{t-1} \rfloor + 1, & i > l_{t-1} - l_t \% l_{t-1}, \end{cases} \tag{6}$$

where $\lfloor \cdot \rfloor$ is a floor function.

In the NA filling manner, the NA decoder fills all boxes at once. The target description probability can be factorized as:

$$P(S|B, R) = \prod_{t=1}^{T} P(w_t|B, R). \tag{7}$$

3.3 Imitating Strategy

Because the SA filling is partially parallel, which fills a box in one step, it better captures the relationship between words and performs better than the NA filling. Therefore, in our model we let the NA filling imitate the SA filling by an online knowledge distillation method [29]. We follow the previous work [12] to make the output of the NA filling close to the SA filling when training them jointly.

In the filling stage, assuming the target words' probabilities of NA filling and SA filling are $\{p^n_{w_1}, p^n_{w_2}, \ldots, p^n_{w_T}\}$ and $\{p^s_{w_1}, p^s_{w_2}, \ldots, p^s_{w_T}\}$ respectively, the imitating loss can be represented by a Kullback-Leibler (KL) Divergence as:

$$\mathcal{L}_{Imit} = \frac{1}{T} \sum_{t=1}^{T} p^n_{w_t} \log \frac{p^n_{w_t}}{p^s_{w_t}}. \tag{8}$$

3.4 Model Training

CE Training Stage. Our BoFiCap model contains two steps in the CE training stage: bounding and filling. The bounding step needs to predict the type of boxes and the number of words to be filled in each box. The objective of this step is to minimize the negative log-likelihood of the correct type of boxes and the number of words using the maximum likelihood estimation (Box Label Loss):

$$\mathcal{L}_{Bound} = -\sum_{t=1}^{N} \log P_\theta(l_t, c_t | b_{<t}, R). \tag{9}$$

The filling step is filling all boxes predicted in advance using NA or SA filling. NA filling fills the boxes in parallel, the loss of which is the sum of the negative log-likelihood of the correct words (NA Generation Loss) as:

$$\mathcal{L}_{NA} = -\sum_{t=1}^{T} \log P_\theta(w_t | B, R). \tag{10}$$

SA filling fills a box in one step, the loss of which can be calculated (SA Generation Loss) as:

$$\mathcal{L}_{SA} = -\sum_{t=1}^{N} \log P_\theta(g_t | g_{<t}, b_{\leq t}, R). \tag{11}$$

We train our BoFicap model jointly in NA and SA filling, so the objective function for BoFiCap can be written as:

$$\mathcal{L} = \mathcal{L}_{Bound} + \mathcal{L}_{NA} + \mathcal{L}_{SA} + \mathcal{L}_{Imit}. \tag{12}$$

RL Training Stage. In the RL training stage, we directly minimize the negative expected reward using SCST [23]:

$$\mathcal{L} = -\frac{1}{M}\sum_{m=1}^{M}(r(S_n) - b) \bigtriangledown \log(p_\theta(S_n)), \tag{13}$$

where r is the CIDEr metric, $p_\theta(S_n)$ is the probability of the m-th sample description, and b is the baseline reward. We refer to the previous work [20] for the value of b and set $M = 5$ in our experiments.

Table 1. Performance comparisons with different evaluation metrics on the test set of MS COCO. All values except Latency and Speedup are reported as a percentage (%). '/' denotes that the results are not reported. '-' denotes unfair comparison because latency is greatly affected by different devices so we do not report them in other models. Our Latency is tested on a GeForce GTX 1080 Ti GPU. The Speedup values are from the corresponding papers. The top results under each decoding manner are in bold. BoFiCap-NA and BoFiCap-SA are our models with the non-autoregressive filling and the semi-autoregressive filling, respectively.

Models	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Latency	Speedup
Autoregressive Models								
AIC (beam = 1)	80.5	38.9	29.0	58.7	129.4	**22.8**	192 ms	**1.73**×
AIC (beam = 3)	**80.9**	**39.3**	29.0	**58.9**	130.2	**22.8**	332 ms	1.00×
Non-Autoregressive Models								
MNIC [9]	75.4	30.9	27.5	55.6	108.1	21.0	-	2.80×
FNIC [6]	/	36.2	27.1	55.3	115.7	20.2	-	8.15×
CMAL [10]	**80.3**	37.3	28.1	58.0	124.0	21.8	-	**13.90**×
IBM [7]	77.2	36.6	27.8	56.2	113.2	20.9	-	3.06×
BoFiCap-NA (*Ours*)	80.1	**38.2**	**28.4**	**58.2**	**125.6**	**22.1**	36 ms	9.22×
Semi-Autoregressive Models								
PNAIC [8]	80.4	38.3	29.0	58.4	**129.4**	22.2	-	2.17×
SATIC [31]	**80.8**	38.4	28.8	58.5	129.0	**22.7**	-	1.65×
SAIC [28]	80.4	38.7	**29.4**	58.5	128.3	22.2	-	1.55×
BoFiCap-SA (*Ours*)	80.5	**38.9**	28.8	**58.8**	128.4	**22.7**	90 ms	**3.69**×

4 Experiments

4.1 Experimental Settings

We experiment with the MS COCO dataset [4] that is widely used in image captioning tasks, including 123,287 images and each image has five captions at least. We refer to the karpathy's split [16] to split the dataset into 113,287, 5,000, and 5,000 images for training, validation, and offline testing. We count all words in captions and omit words that occur less than five times to build a vocabulary that comprises 9,487 words. In addition, for efficiency, we truncate captions that are longer than 16 in training. We use the autoregressive captioning model (AIC) as our teacher model.

Evaluation Metrics. We evaluate our model using a variety of automatic metrics, including BLEU-1/4 [21], METEOR [3], ROUGE [19], CIDEr [26] and SPICE [1], which denoted as B1/4, M, R, C, and S, respectively for convenience. We also use latency and speedup to compare the efficiency to other models.

4.2 Main Comparison Results

We compare the performance of our proposed method to various baseline models in two manners and report the results in Table 1. Our observations are summarized below.

Our proposed method BoFiCap-NA outperforms a variety of compelling non-autoregressive baselines in NA filling, regarding BLEU-4, METEOR, ROUGE, CIDEr, and SPICE metrics, while maintaining a 9.22× speedup. In SA filling, we observe that our proposed method BoFiCap-SA has a competitive performance compared to models in a semi-autoregressive manner, which achieves state-of-the-art results on BLEU-4, ROUGE, and SPICE while maintaining a 3.69× speedup. We report the result of the BoFiCap-SA trained solely since joint training has a negative impact on its performance, and we will discuss this phenomenon in Sect. 5.

5 Analysis

Effect of Hierarchical Split Method. Performance and speed evaluation of BoFiCap models at different hierarchical splits level-k is reported in Table 2. We observe that in SA filling, the acceleration primarily results from the reduction of decoding steps because the model trained with a shallow layer split generates descriptions with fewer boxes. Comparing the results from layers 1 to 2, we see that the speedup changes from 5.19× to 8.51× while only suffering a slight drop in the CIDEr score (1.6 points). In NA filling, the acceleration mainly results from the reduction of bounding box steps and predicting the number of corresponding words because our bounding module bounds boxes in an autoregressive manner. Comparing the results from layers 1 to 2, we see that the speedup changes from 11.45× to 13.83× while only experiencing a slight drop in the CIDEr score (1.7 points).

Ablation Studies. In a non-autoregressive manner, we implement a single vanilla non-autoregressive model named NAIC. In Table 3, we observe that our BoFiCap-NA with bounding boxes greatly improves the CIDEr score from 108.5 to 122.3. Further improvements can be achieved by adding joint training and imitating strategy, resulting in our BoFiCap-NA model reaching the state of the art on the CIDEr score (125.6). Regarding our semi-autoregressive manner, the joint training method results in a slight drop in the performance of our BoFiCap-SA. This is because we use a shared decoder to help our BoFiCap-NA, and their

Table 2. Effect of hierarchical box splits evaluated on MSCOCO test set. k denotes the level of box splits.

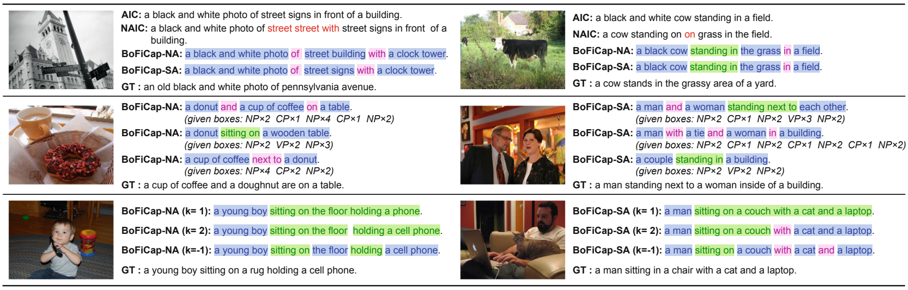
Models	k	B4	M	R	C	S	Speedup
AIC(beam = 3)	/	39.3	29.0	58.9	130.2	22.8	1.00×
BoFiCap-NA	1	36.8	27.9	57.7	122.1	21.5	13.83×
	2	37.6	28.2	58.1	123.8	21.9	11.45×
	−1	38.2	28.4	58.2	125.6	22.1	9.22×
BoFiCap-SA	1	37.6	28.1	58.2	123.3	21.7	8.51×
	2	38.2	28.4	58.4	124.9	22.2	5.19×
	−1	38.5	28.7	58.5	127.5	22.6	3.69×

decoding processes are different. We also observe that the imitating strategy hardly works in BoFiCap-SA because it allows the BoFiCap-NA model to imitate the output of the BoFiCap-SA model, but this strategy has little effect on BoFiCap-SA.

Table 3. Effect of three methods evaluated on MS COCO test set, where Bound, Joint, and Imit represent the bounding boxes, joint training, and Imitating strategy.

Models	Methods	B4	M	R	C	S
NAIC	/	30.8	25.6	55.4	108.5	19.6
BoFiCap-NA	+Bound	37.6	28.1	58.0	122.3	21.8
	+Bound+Joint	37.9	28.3	58.2	124.7	22.0
	+Bound+Joint+Imit	38.2	28.4	58.2	125.6	22.1
BoFiCap-SA	+Bound	38.9	28.8	58.8	128.4	22.7
	+Bound+Joint	38.6	28.7	58.5	127.2	22.7
	+Bound+Joint+Imit	38.5	28.7	58.5	127.5	22.6

Case Study. We present six examples in Fig. 4. In the top two examples shown in Fig. 4, we compare the sentences generated from AIC, NAIC, and two manners of BoFiCap models. Overall, all the models effectively represent the visual content of the given image. Compared to the baseline NAIC, the sentences generated by both manners of our BoFiCap model are fluent and precise. Moreover, our BoFiCap model can accurately assign words with the corresponding box type, resulting in descriptions that are more syntactically structured and less repetitive. To exhibit the diverse generation ability of BoFiCap, we also provide multiple sentences generated with different boxes or split levels in the middle and bottom of Fig. 4.



AIC: a black and white photo of street signs in front of a building.
NAIC: a black and white photo of street street with street signs in front of a building.
BoFiCap-NA: a black and white photo of street building with a clock tower.
BoFiCap-SA: a black and white photo of street signs with a clock tower.
GT : an old black and white photo of pennsylvania avenue.

AIC: a black and white cow standing in a field.
NAIC: a cow standing on on grass in the field.
BoFiCap-NA: a black cow standing in the grass in a field.
BoFiCap-SA: a black cow standing in the grass in a field.
GT : a cow stands in the grassy area of a yard.

BoFiCap-NA: a donut and a cup of coffee on a table.
(given boxes: NP×2 CP×1 NP×4 CP×1 NP×2)
BoFiCap-NA: a donut sitting on a wooden table.
(given boxes: NP×2 VP×2 NP×3)
BoFiCap-NA: a cup of coffee next to a donut.
(given boxes: NP×4 CP×2 NP×2)
GT : a cup of coffee and a doughnut are on a table.

BoFiCap-SA: a man and a woman standing next to each other.
(given boxes: NP×2 CP×1 NP×2 VP×3 NP×2)
BoFiCap-SA: a man with a tie and a woman in a building.
(given boxes: NP×2 CP×1 NP×2 CP×1 NP×2 CP×1 NP×2)
BoFiCap-SA: a couple standing in a building.
(given boxes: NP×2 VP×2 NP×2)
GT : a man standing next to a woman inside of a building.

BoFiCap-NA (k= 1): a young boy sitting on the floor holding a phone.
BoFiCap-NA (k= 2): a young boy sitting on the floor holding a cell phone.
BoFiCap-NA (k=-1): a young boy sitting on the floor holding a cell phone.
GT : a young boy sitting on a rug holding a cell phone.

BoFiCap-SA (k= 1): a man sitting on a couch with a cat and a laptop.
BoFiCap-SA (k= 2): a man sitting on a couch with a cat and a laptop.
BoFiCap-SA (k=-1): a man sitting on a couch with a cat and a laptop.
GT : a man sitting in a chair with a cat and a laptop.

Fig. 4. Examples of captions generated from different models. GT denotes ground-truth captions.

6 Related Work

Non-autoregressive Image Captioning. Overall, non-autoregressive image captioning models can be divided into two categories: latent transformer [15] based and iterative refinement based. Following the latent transformer, Fei [6] firstly predicts ordered keywords with an RNN and then generates the complete sentence simultaneously. Gao *et al.* [9] utilize a refinement strategy on sentence generation, which means the sentence is iterated multiple times and each iteration generates part of the words in the final sentence. Fei [7] also adopts an iterative refinement strategy and constructs a latent variable to bridge the image encoder and textual decoder, which is iteratively optimized to improve generation quality during inference. Besides, Guo *et al.* [10] propose to use multi-agent reinforcement learning to model the sentence-level objective. Different from all the above work, our NA filling method makes two innovations: 1) we use the bounding boxes to enhance the decoder's ability to capture dependencies between words. 2) we improve the performance of the NA filling method by jointly training the model with two filling manners and using the imitating strategy.

Semi-autoregressive Image Captioning. Although the performance of the non-autoregressive model has been enhanced by many methods, it still falls behind of state of the art results [5,14]. Recently, some works have explored how to make a trade-off between quality and speed by utilizing a semi-autoregressive manner. Generally, these methods generate descriptions in a group type introduced in Sect. 2. Zhou *et al.* [31] simply treat each block as one group. Fei [8] organizes the words in the same position in each block into one group. Yan *et al.* [28] adopt a slightly special two-stage generation paradigm, where the first word in each block is generated autoregressively, then the rest words will be generated simultaneously. But our groups are meaningful phrases rather than fixed blocks, and we utilize bounding boxes as hints for each group generation.

7 Conclusion

In this paper, we propose a new framework for image captioning, BoFiCap, that utilizes bounding and filling techniques. In contrast to previous accelerated approaches, our method utilizes the properties of descriptive sentences to decompose generation steps using bounding and filling. Furthermore, our framework provides flexible image description generation to meet the specific needs of users.

Acknowledgement. We would like to thank the anonymous reviewers for their constructive comments. This work was supported by NSFC No. 62176115.

References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: semantic propositional image caption evaluation. In: Proceedings of ECCV (2016)
2. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of CVPR (2018)
3. Banerjee, S., Lavie, A.: Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of ACL workshop (2005)
4. Chen, X., et al.: Microsoft coco captions: data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
5. Cornia, M., Stefanini, M., Baraldi, L., Cucchiara, R.: Meshed-memory transformer for image captioning. In: Proceedings of CVPR (2020)
6. Fei, Z.: Fast image caption generation with position alignment. arXiv preprint arXiv:1912.06365 (2019)
7. Fei, Z.: Iterative back modification for faster image captioning. In: Proceedings of ACM MM (2020)
8. Fei, Z.: Partially non-autoregressive image captioning. In: Proceedings of AAAI (2021)
9. Gao, J., Meng, X., Wang, S., Li, X., Wang, S., Ma, S., Gao, W.: Masked non-autoregressive image captioning. arXiv preprint arXiv:1906.00717 (2019)
10. Guo, L., Liu, J., Zhu, X., He, X., Jiang, J., Lu, H.: Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. arXiv preprint arXiv:2005.04690 (2020)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of CVPR (2016)
12. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR (2015)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. (1997)
14. Huang, L., Wang, W., Chen, J., Wei, X.Y.: Attention on attention for image captioning. In: Proceedings of ICCV (2019)
15. Kaiser, L., et al.: Fast decoding in sequence models using discrete latent variables. In: Proceedings of ICML (2018)
16. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of CVPR (2015)
17. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput, Vis (2017)

18. Li, Y., Pan, Y., Yao, T., Mei, T.: Comprehending and ordering semantics for image captioning. In: Proceedings of CVPR (2022)
19. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out (2004)
20. Luo, R.: A better variant of self-critical sequence training. CoRR (2020)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL (2002)
22. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of NeurIPS (2015)
23. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of CVPR (2017)
24. Song, Z., Zhou, X., Dong, L., Tan, J., Guo, L.: Direction relation transformer for image captioning. In: Proceedings of ACM MM (2021)
25. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)
26. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of CVPR (2015)
27. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of CVPR (2015)
28. Yan, X., Fei, Z., Li, Z., Wang, S., Huang, Q., Tian, Q.: Semi-autoregressive image captioning. In: Proceedings of ACM MM (2021)
29. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of CVPR (2018)
30. Zhang, Y., Zhang, Y., Qi, P., Manning, C.D., Langlotz, C.P.: Biomedical and clinical English model packages in the stanza python NLP library. CoRR (2020)
31. Zhou, Y., Zhang, Y., Hu, Z., Wang, M.: Semi-autoregressive transformer for image captioning. In: Proceedings of ICCV (2021)

Oral: NLP Applications and Text Mining

ASKSpell: Adaptive Surface Knowledge Enhances Tokens' Semantic Representations for Chinese Spelling Check

Xiaobin Lin[1], Jindian Su[1](✉), Xugang Zhou[1], Xiaobin Ye[2], and Dandan Ma[2]

[1] South China University of Technology, Guangzhou, China
sujd@scut.edu.cn
[2] Guangdong Unicomm, Guangzhou, China

Abstract. Chinese Spelling Check (CSC) is a challenging task to detect and correct spelling errors in Chinese texts. Existing state-of-the-art methods try to use pre-trained language model and incorporate phonological or shape information as external knowledge. However, they neglect the importance of key tokens (i.e., misspellings, certain nouns or verbs, etc.) that might have great impacts on the semantics of other tokens and the sentence. They also neither fully utilize the surface knowledge of characters (i.e., their inherent features such as phonetic and shape). In this paper, a novel CSC framework for capturing adaptive surface knowledge (ASKSpell) is proposed, which consists of a PLM-based encoder for building token representations, a knowledge enhancing extractor for extracting surface knowledge, and a knowledge adjusting controller for focusing on key tokens. The extractor utilizes K-layer transformer blocks to extract surface knowledge from input embeddings, which can alleviate the impacts of errors. And the controller employs multi-head attention units to focus on key tokens by regulating each token's surface knowledge which is fed to the backbone. Experiments conducted on three widely used benchmarks demonstrate that our method outperforms current existing models and achieves a new state-of-the-art.

Keywords: Chinese Spelling Check · Surface Knowledge · Key Token

1 Introduction

Chinese Spelling Check (CSC) is an important and challenging task that aims to detect and correct spelling errors occurred in Chinese texts either at word-level or at character-level, as shown in Fig. 1. In the past few years, as the pre-trained languages, e.g., Bert [1], RoBERTa [2] and ELECTRA [3] have developed

The work reported in this paper is supported by the National Science Foundation under Grant No. 61936003, and GuangDong Basic and Applied Basic Research Foundation under Grant No. 2019B151502057.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 485–497, 2023.
https://doi.org/10.1007/978-3-031-44693-1_38

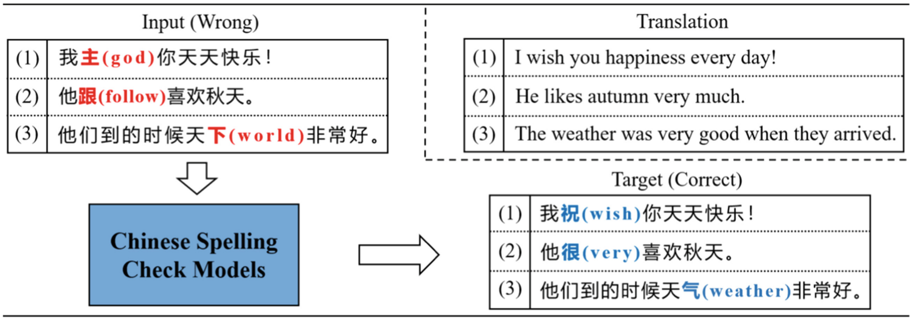

Input (Wrong)
(1) 我主(god)你天天快乐！
(2) 他跟(follow)喜欢秋天。
(3) 他们到的时候天下(world)非常好。
Chinese Spelling Check Models
Translation
(1) I wish you happiness every day!
(2) He likes autumn very much.
(3) The weather was very good when they arrived.
Target (Correct)
(1) 我祝(wish)你天天快乐！
(2) 他很(very)喜欢秋天。
(3) 他们到的时候天气(weather)非常好。

Fig. 1. Examples of the CSC task. The red characters represent misspellings, and the blue characters represent the corrected ones. (1) is a phonological error. (2) is a visual error. (3) is the other type of error. (Color figure online)

rapidly and achieved significant improvements in most natural language processing (NLP) tasks, some CSC methods based on pre-trained language models (PLMs) are proposed, i.e., SM-BERT [4], and gain new state-of-the-art performance on several canonical CSC datasets, i.e., SIGHAN 2013 [5], SIGHAN 2014 [6] and SIGHAN 2015 [7]. Most of other works [8–10] mainly pay attention to how to integrate external phonetic or shape knowledge of characters for enhancing the semantic representations of words and sentences. However, they neglect the importance of those key tokens (i.e., misspellings, certain nouns or verbs, etc.), which might have great impacts on the semantics of other tokens and the sentence. For example, some misspelling tokens might totally disrupt the semantics of other tokens and even change the semantics of the whole sentence [11,12].

In order to better utilize the powerful knowledge of key tokens, this paper provides an automatic framework (ASKSpell) for CSC to capture adaptive surface knowledge (i.e., inherent features of characters, such as their phonetic and shape features.), which enhances tokens' semantic representations, particularly that of key tokens (e.g., misspellings, certain nouns or verbs, etc.), to improve CSC performance without relying on any other external input resources. Specifically, ASKSpell uses PLMs like Bert as the backbone to acquire basic semantics and builds an extractor with a controller. The extractor comprises K-layer transformer blocks that extract the surface features from the original inputs. The controller is composed of multi-head attention units that compute the importance of each token using the backbone's hidden states and input embeddings, and the importance is regarded as the weights of the extractor's outputs. To mitigate the effects of errors, we add the extractor's weighted outputs from each layer to the backbone's last K layers' hidden states in that order. The greater weights there are, the more knowledge from corresponding tokens, especially the counterpart of key tokens, is fed into the backbone. Furthermore, to improve ASKSpell's correction ability, we conduct task-oriented fine-tuning and then use some difficult training instances to guide controller-oriented fine-tuning.

In summary, our contributions are as follows: (1) We are the first one to show that surface knowledge of characters from original inputs, not from external resources, can be better utilized to help reduce the impacts of misspellings for CSC. (2) We propose a novel semantically enhanced ASKSpell that automatically extracts adaptive surface knowledge, especially that of key tokens, to improve CSC performance. ASKSpell is end-to-end trainable and can be easily used in any autoencoder PLMs. (3) We conduct extensive experiments on three CSC benchmarks and achieve new state-of-the-art performance.

2 Related Work

The CSC task focuses on detecting and correcting misspellings and has recently attracted much research. Early works build heuristic rules [13,14] to check for various misspellings. After that, statistical-based methods are frequently employed in this field, such as CSC systems based on the conditional random field [15], the hidden markov model [16], or the n-gram model [17]. Then, neural-based models achieve excellent CSC results, such as RNN-based sequence-to-sequence methods [18,19] and other deep learning methods [8,10] based on PLMs.

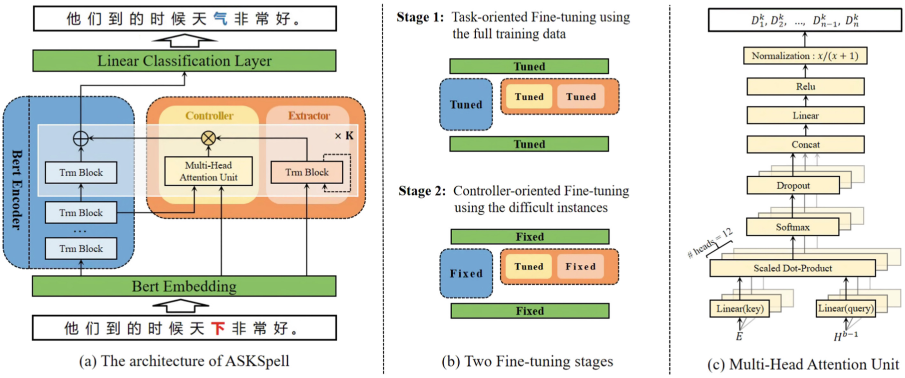
Recently, most works have paid attention to leveraging external knowledge from other operations or resources, such as detection results or character similarity, to enhance the spelling check capability of PLMs. MDCSpell [11] fused the detector's final hidden states into that of the corrector to minimize the misleading impacts of errors. Besides, SpellGCN [8] used a graph convolution network to model the pronunciation and shape similarity of the characters in the confusion set. FASPell [20], REALISE [9], and PLOME [10] integrated different forms of phonetic and visual knowledge to predict correct characters.

However, key tokens, which provide significant context clues [21], are always neglected and errors continue to harm correct characters. In this paper, ASKSpell learns adaptive surface knowledge for each token automatically from training datasets to enhance semantic representations and alleviate the impacts of errors.

3 Approach

3.1 Problem Formulation

The CSC task aims to detect and correct errors in Chinese texts. Generally, given an input sentence $X = \{x_1, x_2, ..., x_n\}$, the CSC model's goal is to detect misspellings and output a target sentence $Y = \{y_1, y_2, ..., y_n\}$, where n denotes the number of characters. Since X and Y have the same length, autoencoder PLMs (e.g., Bert) are natural to be used in this task. We formulate CSC models, like our method, which are based on autoencoder PLMs, as discriminative models. The objective is to maximize the conditional probability $p(Y|X)$.



(a) The architecture of ASKSpell

(b) Two Fine-tuning stages

(c) Multi-Head Attention Unit

Fig. 2. The overall profile of our method. Sub-Figure (a): The red character is an error, and the blue is the corrected one. The dashed arrow indicates that inputs of the next layer are from the front layer, not Bert Embedding. "Trm Block" means transformer block. Sub-Figure (b): "Fixed" and "Tuned" mean that the module's parameters are frozen and tuned, respectively. Sub-Figure (c): All notations are described in Sect. 3.3. "Scaled Dot-Product" is identical to the common implementation [22]. "x" represents each element in the tensor. (Color figure online)

3.2 Motivation and Intuition

The key motivation of ASKSpell is that, in the CSC task, existing methods lack a mechanism to capture key tokens' original features, which are crucial to discover ground truths, and right tokens' semantic representations are affected by errors.

Key tokens have a substantial influence on understanding the context's semantics [21]. Misspellings can be regarded as a kind of key token in the CSC task, and capturing information about misspellings is beneficial to the correction performance [11]. However, other key tokens (e.g., nouns, verbs, etc.) are usually overlooked in previous methods. If we pay more attention to all of these, it will help capture the errors' features and maintain the meanings of other key tokens.

To achieve this, we leverage the power of Bert to comprehend sentence semantics. Then, we use a controller to calculate the tokens' importance in a sentence. Furthermore, we believe that tokens' surface features contain information about themselves, such as phonological or visual features, which can improve the correction, as verified in Sect. 4.5. Therefore, we provide an extractor to capture surface knowledge and integrate it into Bert via the controller to decrease the impact of errors. Intuitively, ASKSpell simulates the human correction process of sentence understanding, key token comprehension, and misspelling analysis to find the ground truth.

3.3 Structure of ASKSpell

The architecture of ASKSpell is shown in Fig. 2(a). The details of each component are described as follows.

Bert Embedding. The Bert embedding generates input representation e_i for a given token, which is the sum of the token, segment, and position embeddings. Because the Bert embedding has become common and our implementation is the same as the original [1], we will omit the detailed description. The token embeddings for a sentence with n characters can be denoted as $E = \{e_1, e_2, ..., e_n\}$, which are the inputs of the Bert encoder, the extractor, and the controller.

Extractor. The extractor is a K-layer transformer encoder (i.e., K transformer blocks) based on the original implementation [22], which contains 12 attention heads, and the hidden state size is 768. Because our implementation is identical to the original and the use of Transformers has become common, we will omit an exhaustive background description of the transformer block. In this work, we denote the sequence of hidden states at the k-th layer of the extractor as $\mathcal{H}^k = \{\mathcal{H}_1^k, \mathcal{H}_2^k, ..., \mathcal{H}_n^k\}$, which can be defined as:

$$\mathcal{H}^k = Trm(\mathcal{H}^{k-1}) \tag{1}$$

where $\mathcal{H}^0 = E$, that means we use the input embeddings as the extractor's inputs, and the extractor captures surface knowledge from the original embedding space. The lower layers of Bert can better integrate surface features [23], so we choose a smaller K to build the extractor. Section 4.5 describes the effects of K in detail.

Controller. The controller consists of K multi-headed attention units, which share parameters in the implementation. The controller helps the extractor capture surface knowledge adaptively and integrate that knowledge into semantic representations better. Specifically, to compute the weights on the extractor's k-th layer's outputs, the multi-head attention unit uses input embeddings E as keys and the b-th layer's hidden states of the Bert encoder H^{b-1} as queries. The operation is shown in detail in Fig. 2(b). We employ a basic normalization strategy $x/(x+1)$ to limit each element of the weights between 0 and 0.5, because our prior experiments show that a wider selection of weights leads to overly enhanced local semantics and reduces the performance. The final weights of each position in the k-th layer are defined as follows:

$$D^k = CtrlUnit(E, H^{b-1}) \tag{2}$$

where $k = b - B + K$, B is the total number of layers of the Bert encoder, and K is the total number of layers of the extractor.

Bert Encoder. We adopt B-layer transformer blocks as the Bert encoder, which is the backbone of ASKSpell. The Bert encoder is the same as the Bert-Base model [1], which means $B = 12$ in our experiments. Specifically, the layers of the Bert encoder from 1 to $B - K$ have the same operation as Formula 1 to obtain hidden states. And then, the adaptive surface knowledge, which is the $\mathcal{H}^k$ of the extractor regulated by the D^k of the controller, is integrated into the upper K

layers' hidden states of the Bert encoder. The sequence of hidden states of the Bert encoder's b-th layer is denoted as $H^b = \{H_1^b, H_2^b, ..., H_n^b\}$. The calculation of the b-th layer can be formulated as:

$$H^b = \begin{cases} Trm(H^{b-1}), & 1 \leq b \leq B - K \\ Trm(H^{b-1}) + D^k \odot \mathcal{H}^k, & B - K < b \leq B \end{cases} \tag{3}$$

where $H^0 = E$, which indicates that we use input embeddings as the Bert encoder's inputs, $k = b - B + K$, and $\odot$ is the element-wise product operation.

Linear Classification Layer. After getting the final Bert encoder's hidden states H^B, we utilize a linear classification layer to predict target characters for each position. The probability distribution of the i-th token is defined as:

$$P(\hat{y}_i|X) = softmax(WH_i^B) \tag{4}$$

where W is a learnable parameter. And the character with the highest probability $argmax_{\hat{y}_i} P(\hat{y}_i|X)$ is viewed as the prediction when inferring.

3.4 Training

The CSC task is defined as the classification task of what correct characters are, and its objective is to minimize the negative log-likelihood of the targets:

$$\mathcal{L} = -\sum_{i=1}^{n} \log P(\hat{y}_i = y_i|X) \tag{5}$$

In practice, after initializing ASKSpell's parameters, we tune them in two stages, as shown in Fig. 2(b). Firstly, we employ full training data to fine-tune all modules of ASKSpell. Secondly, we use the tuned ASKSpell to find difficult instances from full training data that cannot be corrected successfully. Finally, to further adjust the weights of each token's surface features for more complex contexts, we utilize difficult instances to conduct controller-oriented fine-tuning (denoted as CT), which means only tuning the controller.

4 Experiments

In this section, we introduce our experiments and results in detail. Firstly, we present the datasets, evaluation metrics, baselines, and experiment settings. Then we report our main results. Finally, we conduct detailed analyses and discussions to indicate the effectiveness of the extractor and controller.

4.1 Datasets and Metrics

Following previous works [9,20], our full training data in the task-oriented fine-tuning consists of three benchmark datasets [5–7] and the generated pseudo data (denoted as Wang271K) [24]. And the training data in the controller-oriented fine-tuning is the difficult instances, as described in Sect. 3.4. We evaluate our method on SIGHAN13 [5], SIGHAN14 [6], and SIGHAN15 [7] test datasets. We also convert characters in these datasets to simplified Chinese using OpenCC[1]. Table 1 presents the statistic of all datasets.

Table 1. Statistics of datasets. "Err.sent" means erroneous sentence. "Avg.len" means the average length. "Avg.errs" means the average number of misspellings.

Dataset	Training Dataset		Test Dataset	
	#Err.sent/Sent	Avg.len	#Err.sent/Sent	Avg.len
SIGHAN13	340/700	41.8	971/1000	74.3
SIGHAN14	3358/3437	49.6	520/1062	50.0
SIGHAN15	2273/2338	31.3	541/1100	30.6
Wang271K	271009/271329	42.6	–	–

We report the metrics at detection and correction levels, which are commonly used in previous works [9,12,20], and they include accuracy, precision, recall, and F1-score. At the detection level, only all misspellings in a sentence are successfully detected, and the verdict is considered to be right. At the correction level, all misspellings must be detected and corrected to the golden ones. Besides, we run each experiment three times to report the max, mean, and min results.

4.2 Baseline Models

We compare ASKSpell with several typical CSC models to analyze its performance. **SpellGCN** [8] integrates character similarity knowledge into PLMs by a special graph convolution network. **REALISE** [9] utilizes multimodal information of Chinese characters to improve CSC performance. **MDCSpell** [11] uses a detector to capture character features and fuses them with the corrector's hidden states to minimize the impacts of errors. **ECOPO(REALISE)** [12] focuses on negative samples to avoid REALISE predicting common characters. **Bert** [1] uses full training data to fine-tune the Bert-Base model.

4.3 Experiment Settings

The ASKSpell is implemented using the huggingface's PyTorch implementations of transformers [25]. We initialize the weights of the Bert embedding, Bert

[1] https://github.com/BYVoid/OpenCC.

Table 2. The performance of ASKSpell and all baseline models. Note that all baseline results are directly from other published papers. We underline the previous state-of-the-art performance for convenient comparison. The best results are in **bold**. “ASKSpell” means only using task-oriented fine-tuning, and “+ CT” means adding controller-oriented fine-tuning. “− controller” means removing the controller.

Dataset	Approach	Detection-Level				Correction-Level			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
SIGHAN13	SpellGCN [8]	–	80.1	74.4	77.2	–	78.3	72.7	75.4
	REALISE [9]	82.7	88.6	82.5	85.4	81.4	87.2	81.2	84.1
	ECOPO(REALISE) [12]	83.3	**89.3**	83.2	**86.2**	82.1	**88.5**	82.0	85.1
	Bert [9]	77.0	85.0	77.0	80.8	77.4	83.0	75.2	78.9
	ASKSpell	84.3	87.2	84.1	85.6	83.3	86.1	83.1	84.6
	ASKSpell + CT	**84.5**	88.3	**84.2**	**86.2**	**83.6**	87.3	**83.3**	**85.2**
	ASKSpell - controller	82.9	87.1	82.6	84.8	81.7	85.8	81.4	83.5
SIGHAN14	SpellGCN [8]	–	65.1	69.5	67.2	–	63.1	67.2	65.3
	REALISE [9]	78.4	67.8	71.5	69.6	77.7	66.3	70.0	68.1
	MDCSpell [11]	–	70.2	68.8	69.5	–	69.0	67.7	68.3
	ECOPO(REALISE) [12]	79.0	68.8	72.1	70.4	78.5	67.5	71.0	69.2
	Bert [9]	75.7	64.5	68.6	66.5	74.6	62.4	66.3	64.3
	ASKSpell	82.2	70.4	74.5	72.4	81.3	68.6	72.6	70.5
	ASKSpell + CT	**82.5**	**71.0**	**74.9**	**72.9**	**81.6**	**69.1**	**73.0**	**71.0**
	ASKSpell - controller	80.6	68.1	72.8	70.4	79.7	66.3	70.8	68.5
SIGHAN15	SpellGCN [8]	–	74.8	80.7	77.7	–	72.1	77.7	75.9
	REALISE [9]	84.7	77.3	81.3	79.3	84.0	75.9	79.9	77.8
	MDCSpell [11]	–	**80.8**	80.6	80.7	–	**78.4**	78.2	78.3
	ECOPO(REALISE) [12]	85.0	77.5	82.6	80.0	84.2	76.1	81.2	78.5
	Bert [9]	82.4	74.2	78.0	76.1	81.0	71.6	75.3	73.4
	ASKSpell	86.7	78.2	83.0	80.5	85.8	76.4	81.0	78.6
	ASKSpell + CT	**86.8**	78.6	**83.2**	**80.8**	**86.0**	76.9	**81.4**	**79.1**
	ASKSpell - controller	85.8	76.8	82.2	79.4	84.3	73.6	78.9	76.1

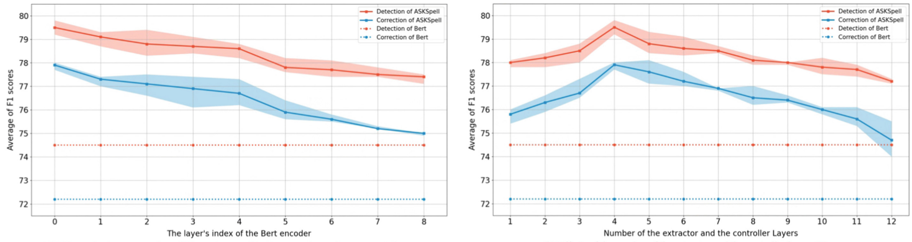
encoder, and linear classification layer by the Bert-Base-Chinese[2], which is a pre-trained masked language model. Besides, we set the number of layers of the extractor and controller as K = 4 by default. We initialize the extractor by the weights of the Bert-Base-Chinese lowest four layers and use the corresponding parameters of its first layers to initialize the controller's linear (query) and linear (key). For all experiments, we only use Adamw [26] optimizer with learning rate warming up and exponential decay, and set epoch size as 2, batch size of training or test as 16, basic learning rate as 5e−5, and max sentence length as 128.

Following previous works [9,12], to accurately reflect the real model performance on the SIGHAN13 test dataset, we also remove all the detected and corrected “的 ”, “地 ”, and “得 ” characters from the model outputs and then evaluate with the ground truth.

[2] https://huggingface.co/bert-base-chinese.

4.4 Main Results

As shown in Table 2, ASKSpell outperforms all baseline models on the SIGHAN14 and SIGHAN15 test datasets, and ASKSpell+CT achieves a new state-of-the-art on all test datasets, which verifies the effectiveness of our framework and training strategy. We observe that our method outperforms MDCSpell on recall and F1 scores, which indicates that our method is better at fusing the capturing knowledge than MDCSpell. Although MDCSpell incorporates features of misspellings, it depends on external detection information and only focuses on errors, not most of key tokens. Besides, our method gets higher F1 scores than ECOPO(REALISE), because REALISE fuses character similarity merely and ECOPO could not mitigate the effects of errors. Finally, ASKSpell and ASKSpell+CT beat vanilla Bert in all metrics on three datasets. This indicates that incorporating adaptive surface knowledge, particularly the counterpart of key tokens, into the semantic representations of tokens is valid for CSC.



(a) Effects of using outputs from different layers of the Bert encoder as the extractor's inputs

(b) Effects of the number of the extractor and the controller layers (K)

Fig. 3. Effects of different input spaces and the number of layers for the extractor and the controller.

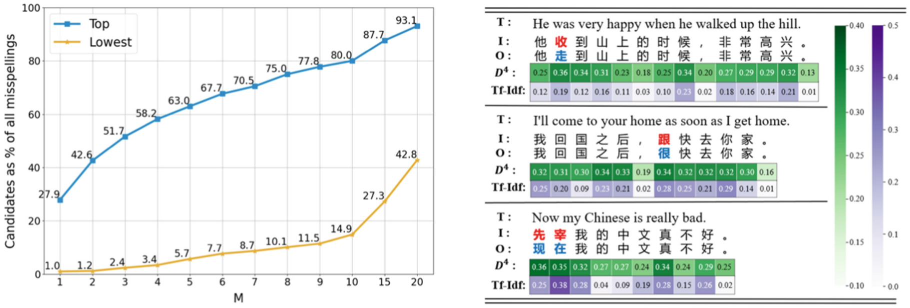
We carry out ablation tests to study the effectiveness of the controller. Table 2 shows that ASKSpell-controller outperforms vanilla Bert but falls short of ASKSpell. It indicates that the knowledge captured by the extractor is useful for improving Bert's results, and the controller is critical in fusing the relevant knowledge into semantic representations.

4.5 Analysis and Discussion

In this section, we verify the effectiveness of surface knowledge captured by the extractor and the controller. After that, outputs of the controller are shown to confirm its concentration on key tokens, especially misspellings.

Effectiveness of Surface Knowledge Captured by the Extractor and Controller. We conduct several ablation tests on three test datasets to check if the surface knowledge captured by the extractor is valuable, and we report their average F1 scores. Firstly, we employ the hidden states H^b of the Bert encoder's b-th layer as the extractor's inputs, where b ranges from 0 to 8. As shown in Fig. 3(a), the H^b with the higher b has lower detection and correction performance. Based on previous research [23], we can conclude that H^b with a lower b has more surface features but less semantic knowledge. Therefore, it indicates that the CSC results improve when the extractor captures more surface knowledge from its inputs.

Secondly, we analyze the effects of the number of extractor layers K. Figure 3(b) shows that the average detection or correction F1 scores climb initially and then decrease as K increases, and $K = 4$ achieves the best results. This indicates that the extractor requires multiple layers to keep functioning. However, too many layers result in H^b and $\mathcal{H}^{b-B+K}$ capturing similar knowledge, which leaves them unable to integrate evident surface knowledge but just the similar into semantic representations. When $K = 4$, the extractor and the controller can fit the surface knowledge well and incorporate it into Bert's semantic representations.



(a) The percentage of candidate errors to the total misspellings of all samples in the SIGHAN15 test dataset, where the D^4 values of candidates are in the Top or Lowest M of all characters for each sample.

T : He was very happy when he walked up the hill.

I :	他	收	到	山	上	的	时	候	，	非	常	高	兴	。
O :	他	走	到	山	上	的	时	候	，	非	常	高	兴	。
D^4 :	0.25	0.36	0.34	0.31	0.23	0.18	0.25	0.34	0.20	0.27	0.29	0.29	0.32	0.13
Tf-Idf:	0.12	0.19	0.12	0.16	0.11	0.03	0.10	0.23	0.02	0.18	0.16	0.14	0.21	0.01

T : I'll come to your home as soon as I get home.

I :	我	回	国	之	后	，	跟	快	去	你	家	。
O :	我	回	国	之	后	，	很	快	去	你	家	。
D^4 :	0.32	0.31	0.30	0.34	0.33	0.19	0.34	0.32	0.32	0.32	0.30	0.16
Tf-Idf:	0.25	0.20	0.09	0.23	0.21	0.02	0.28	0.25	0.21	0.29	0.14	0.01

T : Now my Chinese is really bad.

I :	先	宰	我	的	中	文	真	不	好	。
O :	现	在	我	的	中	文	真	不	好	。
D^4 :	0.36	0.35	0.32	0.27	0.27	0.24	0.34	0.24	0.29	0.25
Tf-Idf:	0.25	0.38	0.28	0.04	0.09	0.19	0.28	0.15	0.26	0.02

(b) Each token's importance visualization. "T" is the translation. "I" is the input. "O" is the output. D^4 is the 4-th layer's output of the controller. "Tf-Idf" is a statistical method to evaluate the importance of a token in a sentence.

Fig. 4. Statistics of the degree of focusing on misspellings and each token's importance visualization.

Visualization of the Controller's Outputs. We select the misspellings, whose D^4 values (the fourth layer's outputs of the controller) are in the Top or Lowest M of all characters for each sample, as candidates. And then we calculate the percentage of candidates to the total misspellings of all samples in the SIGHAN15 test dataset. Figure 4(a) shows that 27.9% of misspellings have the largest D^4 in the samples to which they belong, only 1.0% have the lowest D^4, and those in the top 10 D^4 have achieved 80.0%. That is, the weights of the extractor's outputs of most errors are larger when fusing the outputs into the Bert encoder, which means that the controller can focus on most misspellings.

After that, we visualize the D^4 of three samples to show the controller's outputs for analyzing the weighting mechanism of the extractor's outputs. And we employ a traditional statistical method called term frequency-inverse document frequency (Tf-Idf) [27] on the wiki2019zh (i.e. A general corpus containing one million pages from Wikipedia dump as of February 7, 2019) corpus and the input sentence to determine the importance of each character. Figure 4(b) shows that misspellings have higher D^4 in each sample, and the distributions of D^4 and Tf-Idf are approximate, i.e., regions with higher Tf-Idf values have higher D^4 values, whereas positions with lower Tf-Idf values have lower D^4 values. Besides, we find an increasing Pearson correlation (reaching 0.642) between D^4 and Tf-Idf during training, which suggests the high relativeness between them. This indicates that the controller emphasizes important characters, which is consistent with the controller's mission.

5 Conclusion

We are the first one to prove that surface knowledge can benefit the CSC task, and then propose the ASKSpell, which captures adaptive surface knowledge from training data and focuses on key tokens, to enhance semantic representations and reduce the impacts of errors. The empirical comparison and the results of analytical studies verify its effectiveness. Because ASKSpell only learns from training data and the Bert and extractor can integrate various information effectively, we believe ASKSpell will benefit a broader range of application scenarios that require surface knowledge and semantic representations of characters, not only the CSC task. We leave this direction to future work.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the NAACL, Minneapolis, Minnesota, pp. 4171–4186. ACL (2019)
2. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
3. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
4. Zhang, S., Huang, H., Liu, J., Li, H.: Spelling error correction with soft-masked BERT. In: Proceedings of the 58th Annual Meeting of the ACL, pp. 882–890. ACL, Online (2020)
5. Wu, S.H., Liu, C.L., Lee, L.H.: Chinese spelling check evaluation at SIGHAN bake-off 2013. In: Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, pp. 35–42. Asian Federation of NLP (2013)
6. Yu, L.C., Lee, L.H., Tseng, Y.H., Chen, H.H.: Overview of SIGHAN 2014 bake-off for Chinese spelling check. In: Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, pp. 126–132. ACL (2014)
7. Tseng, Y.H., Lee, L.H., Chang, L.P., Chen, H.H.: Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp. 32–37. ACL (2015)

8. Cheng, X., et al.: SpellGCN: incorporating phonological and visual similarities into language models for Chinese spelling check. In: Proceedings of the 58th Annual Meeting of the ACL, pp. 871–881. ACL, Online (2020)
9. Xu, H.D., et al.: Read, listen, and see: leveraging multimodal information helps Chinese spell checking. In: Findings of the ACL, pp. 716–728. ACL, Online (2021)
10. Liu, S., Yang, T., Yue, T., Zhang, F., Wang, D.: PLOME: pre-training with misspelled knowledge for Chinese spelling correction. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on Natural Language Processing, pp. 2991–3000. ACL, Online (2021)
11. Zhu, C., Ying, Z., Zhang, B., Mao, F.: MDCSpell: a multi-task detector-corrector framework for Chinese spelling correction. In: Findings of the ACL, Dublin, Ireland, pp. 1244–1253. ACL (2022)
12. Li, Y., et al.: The past mistake is the future wisdom: error-driven contrastive probability optimization for Chinese spell checking. CoRR (2022)
13. Jiang, Y., et al.: A rule based Chinese spelling and grammar detection system utility. In: 2012 International Conference on System Science and Engineering, pp. 437–440 (2012)
14. Chu, W.C., Lin, C.J.: NTOU Chinese spelling check system in Sighan-8 bake-off. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp. 137–143. ACL (2015)
15. Wang, Y., Liao, Y.: NCTU and NTUT's entry to CLP-2014 Chinese spelling check evaluation. In: Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, pp. 216–219. ACL (2014)
16. Zhang, S., Xiong, J., Hou, J., Zhang, Q., Cheng, X.: HANSpeller++: a unified framework for Chinese spelling correction. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp. 38–45. ACL (2015)
17. Xie, W., et al.: Chinese spelling check system based on n-gram model. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp. 128–136. ACL (2015)
18. Yang, Y., Xie, P., Tao, J., Xu, G., Li, L., Si, L.: Alibaba at IJCNLP-2017 task 1: embedding grammatical features into LSTMs for Chinese grammatical error diagnosis task. In: Proceedings of the IJCNLP 2017, Shared Tasks, pp. 41–46, Taipei, Taiwan. Asian Federation of Natural Language Processing (2017)
19. Qiu, Z., Qu, Y.: A two-stage model for Chinese grammatical error correction. IEEE Access, 146772–146777 (2019)
20. Hong, Y., Yu, X., He, N., Liu, N., Liu, J.: FASPell: a fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm. In: Proceedings of the 5th W-NUT, Hong Kong, China, pp. 160–169. ACL (2019)
21. Li, H., Zhu, J., Zhang, J., Zong, C., He, X.: Keywords-guided abstractive sentence summarization. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA, pp. 8196–8203. AAAI Press (2020)
22. Vaswani, A., et al.: Attention Is All You Need. arXiv e-prints p. 1706.03762 (2017)
23. Jawahar, G., Sagot, B., Seddah, D.: What does BERT learn about the structure of language? In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the ACL, Florence, Italy, pp. 3651–3657. ACL (2019)
24. Wang, D., Song, Y., Li, J., Han, J., Zhang, H.: A hybrid approach to automatic corpus generation for Chinese spelling check. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on EMNLP, Brussels, Belgium, pp. 2517–2527. ACL (2018)
25. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on EMNLP, pp. 38–45. Online (2020)

26. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in Adam. CoRR (2017)
27. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on EMNLP, pp. 216–223 (2003)

Biomedical Entity Normalization Using Encoder Regularization and Dynamic Ranking Mechanism

Siye Chen, Chunmei Xie, Hang Wang, Shihan Ma, Yarong Liu, Qiuhui Shi,
Wenkang Huang, and Hongbin Wang(✉)

Ant Group, Hangzhou, China
{chensiye.csy,hongbin.whb}@antgroup.com

Abstract. Biomedical entity normalization is a fundamental method for lots of downstream applications. Due to the rich additional information for biomedical entities in medical dictionaries, such as synonyms or definitions, transformer-based models are applied to dig semantic representations in normalization recently. Despite the high performance of the transformer-based model, the over-fitting problem remains challenging and unsolved. Besides, bi-encoder structure and cross-encoder structure are popularly applied in many biomedical entity normalization works, the issue to measure the distance of such encoder structures is very challenging. Moreover, the triples margin ranking loss mechanism is widely used in reranking stage of entity normalization. In this paper, we proposed an encoder-level regularization to restrain the over-fitting problem caused by the deep representation of transformer. Moreover, we use a dynamic margin ranking mechanism instead of fixed margin selection in reranking stage during training. In this way, we experiment our model on three biomedical entity normalization datasets, and the empirical results outperform previous state-of-the-art models.

1 Introduction

Biomedical entity normalization is the task of mapping mentions in text (e.g. a PubMed article) to a known standard identifiers listed in a reference knowledge dictionary. This kind of task is widely used in many applications like biomedical literature mining [1]. The performance of downstream tasks, such as [2,3], have strong correlation with the normalization task. A standard identifier in a biomedical knowledge dictionary usually provides rich information, which is always encoded in pre-trained language models such as BERT [4] to learn better representations of mentions and medical knowledge dictionary. When training such a deep model encoded by BERT, regularization techniques [5] are essential to prevent overfitting and improve the generalization ability. Besides, biomedical entity normalization task always follows a *candidates generator* and *rerank* paradigm, and neural reranking method [6,7] is widely applied in reranking stage. These neural rankers focus on the similarity of mention and entity but ignore the distance information between them.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 498–510, 2023.
https://doi.org/10.1007/978-3-031-44693-1_39

In this paper, we proposed a *encoder-level regularization* method to restrain overfitting when training and a *dynamic ranking mechanism* to magnify the mention-entity relevant score for biomedical entity normalization. Concretely, we use two encoding architectures, called bi-encoder and cross-encoder, to regularize the model performance when training on the deep model encoded by BERT. Since the bi-encoder faces the upper boundary of representation capacity according to theoretical analysis in [8] due to the limitation of single vector representation, while cross-encoder are computationally expensive and impractical for candidates generator stage, we consider minimizing the distribution distance between bi-encoder and cross-encoder. Besides, we apply a dynamic ranking mechanism, which changes the margin as the in-batch scores changes, to broaden the distance between positive and negative mention-entity pairs relevance scores, which are similarly computed as relevance scores used in [6,9,10]. We evaluate our model on three biomedical entity normalization datasets and achieve new state-of-the-art performance on all datasets, which leads to an improvement of 0.16%–0.73% on top1 accuracy. Further analysis shows the effectiveness of encoder-level regularization and dynamic ranking mechanism.

We summarize our contributions as follows:

1 We proposed a novel method for the biomedical entity normalization task, which uses the encoder-level regularization to restrain the overfitting of the deep model and a dynamic ranking mechanism to enhance the mention-entity pairs.
2 We theoretically show that our encoder-level regularization reduces the inconsistency between training and inference of reranking stage and the effectiveness of the dynamic ranking mechanism.
3 We evaluate our proposed model on three public datasets, namely the NCBI-Disease, BC5CDR-Disease, and BC5CDR-chemical datasets. Extensive experiments show a state-of-the-art result in biomedical entity normalization

2 Related Work

2.1 Biomedical Entity Normalization

The task of biomedical entity normalization is to map an input mention to a known standard identifier in the knowledge dictionary. There existing many biomedical normalization methods and traditional works. Recent approaches applied machine learning techniques, such as DNorm [2] use tf-idf to learn similarities between mention and entity, CNN-ranking [11] use a convolutional neural network to learn representations, NSEEN [12], which use a siamese recurrent neural to learn the similarity between mention and entity. With the advent of deep language model, contextual representation learning for mention and entity become more popular. There are some BERT-based biomedical normalization works. BioBert [13] is a pre-trained on biomedical corpora using BERT [4]. BioSyn [9] uses synonym marginalization for biomedical entity representations. BCNH [10] reformulate the entity candidates by leveraging hypernyms.

[7,14] leverage semantic type information for improved entity normalization. Our model also uses pre-trained BioBERT for learning mention and entity representations.

2.2 BERT-Based Encoder Reranking Architecture

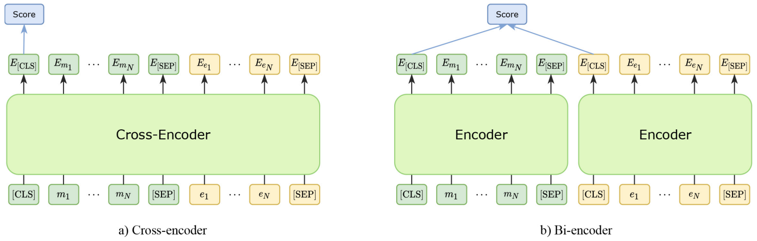

Score
$E_{[CLS]}$ E_{m_1} ··· E_{m_N} $E_{[SEP]}$ E_{e_1} ··· E_{e_N} $E_{[SEP]}$
Cross-Encoder
[CLS] m_1 ··· m_N [SEP] e_1 ··· e_N [SEP]
a) Cross-encoder
Score
$E_{[CLS]}$ E_{m_1} ··· E_{m_N} $E_{[SEP]}$ $E_{[CLS]}$ E_{e_1} ··· E_{e_N} $E_{[SEP]}$
Encoder
Encoder
[CLS] m_1 ··· m_N [SEP] [CLS] e_1 ··· e_N [SEP]
b) Bi-encoder

Fig. 1. The different architecture of encoder type

With the development of deep language model, fine-tuned deep pre-trained BERT achieve advanced re-ranking performance [9,10,13]. Figure 1(a) show the cross-encoder based reranker. It feeds the concatenation of mention and entity text to BERT encoder and takes the *[CLS]* token's embeddings to generate a relevance score. Since mention-entity pairs need computationally expensive cross-attention operations [15], so it is impractical to apply in candidate generator stage and is usually deployed in the reranking stage.

As for candidates generator stage, bi-encoder is the mostly adopted architecture [16] for it can be easily and efficiently employed with support from the approximate nearest neighbor (ANN) [17]. As illustrated in Fig. 1(b), mention and entity are individually fed into the encoder and generate single vector representations respectively, and the relevance score is computed by the similarity of their embeddings. [9,10] used inner production to measure the similarity score.

In our framework, bi-encoder is deployed in the candidates generator stage for better retrieval, and inner production is also adopted to measure the similarity of mention-entity pair. In reranking stage, cross-encoder is applied to do the encoder-level regularization, which can learn the contextualized representations and restrain the overfitting of the model.

3 Methodology

3.1 Task Definition

Given a mention m as an entity string in the biomedical corpus. Each mention m has its own CUI (Concept Unique ID) c in biomedical knowledge base C, such as

MEDIC [18] and Comparative Toxicogenomics Database (CTD) [19]. Our goal is to link the mention m to the ground-truth CUI c^* as follows:

$$c^* = \mathsf{CUI}(\text{argmax}_{c_i \in C}(P(c_i|m;\theta))$$

where $\mathsf{CUI}(\cdot)$ is the standard CUI of mention m and θ is the trainable parameter of our model.

3.2 Method Details

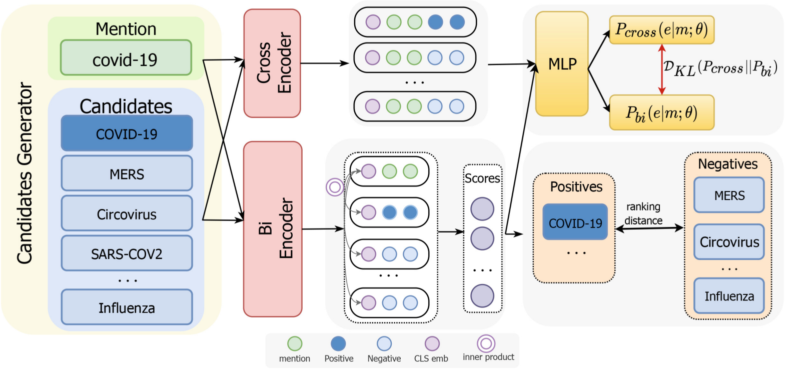


Fig. 2. The architecture of the proposed model. Candidates Generator produces candidates, and feeds into bi-encoder and cross-encoder to get the representations respectively. The KL divergence $\mathcal{D}_{KL}$ of P_{cross} and P_{bi} is adopted for encoder regularization, and dynamic ranking mechanism is applied to evaluate the distance between positives and negatives.

The architecture of our framework is illustrated in Fig. 2. Our approach consists of three parts: 1) Candidates Generator is reused as shown in [9] to generate the entity candidates from entity knowledge base. 2) Encoder level regularization is applied to restrain overfitting in the reranking stage. 3) Dynamic ranking mechanism is deployed to separate the gold entity from other candidates.

Candidates Generator. We adopt the candidates generator method from BioSyn [9]. For a given mention m and a set of entities dictionary D, our goal is to generate candidate entities E of the mention. Sparse encoder, which is built with tf-idf, and dense encoder, which is build with BioBert [13], will firstly be applied to encode mention m and each entity e_i in dictionary $D = \{e_1, e_2, \cdots\}$. The sparse representations of mention m and entity e_i are denoted as v^s_m and $v^s_{e_i}$. The special *[CLS]* token is selected to get the dense representations of mention m and each entity e_i, which are denoted as v^d_m and $v^d_{e_i}$.

Candidates will be generated based on the similarity score of mention m and each entity e_i. The sparse similarity score denoted as $S_{sparse}(m, e_i)$ and dense similarity score denoted as $S_{dense}(m, e_i)$ are hybrid calculated to indicate the entity candidates. Similarity scores are calculated as follows:

$$S_{sparse}(m, e_i) = f(v_m^s, v_{e_i}^s) \tag{1}$$

$$S_{dense}(m, e_i) = f(v_m^d, v_{e_i}^d) \tag{2}$$

$$S(m, e_i) = S_{dense}(m, e_i) + \lambda S_{sparse}(m, e_i) \tag{3}$$

where function $f(\cdot)$ is the inner product of two vectors and λ is a trainable scalar weight for the sparse score. After the similarity score has been calculated, entity candidates $\mathcal{E}_m = [e_1, e_2, \cdots, e_k]$ will be generated due to the top highest k score $\mathcal{S}_m = [s_1, s_2, \cdots, s_k]$.

The entity candidates are pre-computed and will be updated iteratively every step in training stage, while the top candidate entity e^* is retrieved using the similarity score S calculated between mention m and entity dictionary D in inference stage. A conventional contrastive-learning loss is applied to learn the similarity of mention and entity candidates.

Contrastive Estimation. Given a mention m, and a set of entity candidates $\mathcal{E} = [e_1, e_2, \cdots, e_k]$ generated by Candidates Generator, the probability of each entity e is obtained as:

$$P(e|m;\theta) = \frac{\exp(S(m,e))}{\sum\limits_{e_i \in \mathcal{E}} S(m, e_i)} \tag{4}$$

We use a conventional contrastive-learning loss to learn the similarity between mention and entity candidates.

$$\mathcal{L}_{CE} = -\log \frac{\exp(S(m,e))}{\exp(S(m,e)) + \sum\limits_{e' \in \mathcal{E}_N} \exp(S(m,e'))} \tag{5}$$

where $\mathcal{E}_N \in \mathcal{E}$ is a set of negative candidates that exclude all positive candidates in $\mathcal{E}$. [20–22] prove that the hard negatives, which are generated by the hybrid candidates generator, are beneficial for entity representation and entity retrieval.

Encoder-Level Regularization. Figure 1 shows the different structure of mention and entity encoder method. Inspired by Rdrop [5], we proposed an encoder-level Regularization for reranking stage.

Given a mention m and entity $e_i \in \mathcal{E}_m = [e_1, e_2, \cdots, e_k]$, which are top k entity candidates of mention m. For the bi-encoder structure, we reuse the formula (4) to obtain the probability of each entity, which denotes as $P_{bi}(e|m;\theta)$. For cross-encoder structure, we get a cross probability score as follows:

$$v_{cross} = \mathbf{enc}([\mathrm{CLS}]m[\mathrm{SEP}]e_i[\mathrm{SEP})_{[\mathrm{CLS}]} \tag{6}$$

$$S_{cross}(m, e) = \mathbf{MLP}(v_{cross}) \tag{7}$$

$$P_{cross}(e|m;\theta) = \frac{\exp(S_{cross}(m, e))}{\sum_{e_i \in \mathcal{E}} S_{cross}(m, e_i)} \tag{8}$$

where **enc** is the BioBERT pretrained model and **MLP** is the multi-layer perceptron (MLP). Similar to candidates generator, the special *[CLS]* token is selected to get the dense representations.

We then minimize bidirectional Kullback-Leibler (KL) divergence between the two probability distributions P_{bi} and P_{cross} to regularize the model similarity calculation.

$$\mathcal{L}_{KL} = \frac{1}{2}(\mathcal{D}_{KL}(P_{bi}||P_{cross}) + \mathcal{D}_{KL}(P_{cross}||P_{bi})) \tag{9}$$

where $\mathcal{D}_{KL}(P_1||P_2)$ is the Kullback-Leibler (KL) divergence between two probability distributions P_1 and P_2.

Our intuition is that, by enforcing biomedical normalization to train persistently with different encoder architecture, our model can reduce the inconsistency between training and inference. In this way, our encoder-level reranking regularization further regularizes the similarity between mention and entity beyond the encoder method and improves the generalization ability reranking model.

Dynamic Ranking Mechanism. Given a mention m and entity candidates $\mathcal{E}_m = [e_1, e_2, \cdots, e_k]$, the standard pairwise hinge ranking loss defined on a triple (m, e^+, e^-) as follows:

$$\mathcal{L}(m, e^+, e^-) = \max(0, \lambda - S(m, e^+) + S(m, e^-)) \tag{10}$$

where e^+ is the positive sample in entity candidates while e^- is the negative sample in entity candidates, λ is the margin for the ranking loss.

Note that each instance's score calculated by inner product differs from other instances, then the scalar margin λ need to be adaptive to the in-batch instances. Meanwhile, each mention includes multi positives entity candidates and multi negative entity candidates, building such triple (m, e^+, e^-) has high complexity. Therefore, we modified the triple margin loss as follows:

$$\mathcal{L}_{rank}(m, \mathcal{E}^+, \mathcal{E}^-) = \max(0, \hat{\lambda} - \min(S(m, \mathcal{E}^+)) + \max(S(m, \mathcal{E}^-))) \tag{11}$$

$$\hat{\lambda} = \max(0, \mathrm{mean}(S(m, \mathcal{E}^+)) - \mathrm{mean}(S(m, \mathcal{E}^-))) \cdot \mu \tag{12}$$

where $S(m, \mathcal{E}^+)$ are scores for all positive set $\mathcal{E}^+ \in \mathcal{E}$ while $S(m, \mathcal{E}^-)$ are scores for all negative set $\mathcal{E}^- \in \mathcal{E}$, μ is a scalar value to adjust margin loss. Our intuition is that, using $\min(S(m, \mathcal{E}^+))$ and $\max(S(m, \mathcal{E}^-))$ to ensure that the positives pairs are always bigger than the negatives pairs in an easy way. In this way, the margin $\hat{\lambda}$ can be adaptive to the in-batch samples.

Combined Loss. For the framework, our purpose is to get top entity e for given mention m. By improving the effect and robust similarity score of m and e, we introduce two hyper-parameters to sum Eq. 5, Eq. 9, Eq. 11. The overall loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \cdot \mathcal{L}_{KL} + \beta \cdot \mathcal{L}_{rank} \tag{13}$$

where α and β are hyper-parameters for balanced training. Our main experiments use a simple but effective combination method which consists in given equal α and β.

Table 1. Overall statistics of the three datasets.

Dataset	#documents			#mentions		
	train	dev	test	train	dev	test
NCBI-disease	592	100	100	5,134	787	960
BC5CDR-disease	500	500	500	4,182	4,244	4,424
BC5CDR-chemical	500	500	500	5,203	5,347	5,385

4 Experiment

4.1 Datasets and Experiment Setup

We use three available public datasets in this experiment, namely NCBI-disease - the NCBI disease corpus [23] and BC5CDR-disease, BC5CDR-chemical - the BioCreative V CDR task corpus [24]. Table 1 presents the detailed statistical information of the three datasets.

NCBI-Disease. NCBI Disease dataset [23][1] contains 792 PubMed documents, which was split into 692 documents for training and development, and 100 documents for testing. Mention in each PubMed document was manually annotated with its mapping CUI in the MEDIC dictionary [18]. In our experiments, we use the Dec 1, 2021 version of the MEDIC disease dictionary containing 13144 CUIs and 74592 synonyms.

Biocreative V CDR. BioCreative V CDR [24][2] contains 1,500 PubMed documents, which are equally split into three parts for training, development, and test, respectively. It provides disease and chemical type entities, which are manually annotated in MEDIC dictionary and Comparative Toxicogenomics Database (CTD) chemical dictionary [19]. In our experiments, we use the Dec 1, 2021 version of the CTD chemical dictionary containing 174,822 CUIs and 272,366 synonyms.

Table 2. Overall comparisons of different models for biomedical entity normalization on three datasets.

Models	NCBI-Disease		BC5CDR-Disease		BC5CDR-Chemical	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
Sieve-Based [25]	84.7	–	84.1	–	90.7	–
Taggerone [31]	87.7	–	88.9	–	94.1	–
CNN Ranking [11]	86.1	–	–	–	–	–
NormCo [26]	87.8	–	88.0	–	90.7	–
BNE [27]	87.7	–	90.6	–	95.8	–
BERT Ranking [28]	89.1	–	–	–	–	–
TripletNet [29]	90.0	–	–	–	–	–
BioSyn [9] ♠	89.08	94.24	92.62	96.01	96.27	97.26
BCNH [10]	90.61	95.00	–	–	–	–
Ours ♠	**90.72**	**95.46**	**93.35**	**96.48**	**96.45**	**97.50**

♠ Due to the different versions of MEDIC and CTD dictionaries, we run the open-source code 10 times with different seeds and evaluate the results with 95% confidence. The results of other models are directly cited.

We apply some same preprocessing steps for each mention and each entity as in BioSyn [9]. 1) *Spelling Check*; 2) *Abbreviation Resolution*; 3) *Composite mention split*;

In the candidates generator stage, we set all parameters exactly the same with BioSyn [9]. Parameters like max length, number of top candidates, dense ratio α for candidates retrieval, learning rate, weight decay mini-batch size, epochs are set to 25, 20, 0.5, 1e−5, 1e−2, 16, 10 respectively. Our model only introduces a new hyperparameter $\mu = 0.5$, which scales the dynamic ranking margin λ.

Following previous biomedical entity normalization [9–11,25–29], we use the top k accuracy as an evaluation metric. In our experiments, we use Acc@1 to evaluate top 1 accuracy while Acc@5 to evaluate top 5 accuracy respectively. The AdamW [30] optimizer is applied for parameter optimization.

4.2 Experiment Results

We compare our model with the baseline methods on the biomedical entity normalization. The detailed experiment results are shown in Table 2. Our model outperforms all previous models on the three datasets and achieves new state-of-the-art performance. Our model gained 0.16% and 0.46% performance of Acc@1 and Acc@5 on NCBI-Disease. Besides, our model also gained 0.73% and 0.18% performance of Acc@1 on BC5CDR-Disease and BC5CDR-Chemical respectively.

[1] https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE.

[2] https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr.

4.3 Discussion

To dig into our model improvement beyond with BioSyn, the ablation study is conducted to present the effectiveness of encoder-level reranking regularization and dynamic ranking mechanisms proposed in the model. The analysis of encoder-level regularization is also applied to prove the effectiveness of the component. Besides, we also study the influence of different ranking strategies.

Ablation Study. Table 3 presents the effectiveness of the encoder-level regularization (Enc-Reg) and dynamic ranking loss mechanism (DRL). The first experiment reports the results of our model without Enc-Reg. Though the Enc-Reg component improves not much, it still gains better results due to the regularization mechanism. The second experiment reports the results of our model without DRL. We can know that, when removed the DRL component, our model achieves worse results on three datasets. It proves that the DRL can dynamically widen the distance between positive candidates and negative samples when training. The third experiment reports the results for our model with both encoder-level regularization and dynamic ranking loss mechanism. The ablation study proves the importance and effectiveness of the two components to our model.

Table 3. Ablation Study for each component.

Models	NCBI-D.		BC5CDR-D.		BC5CDR-C.	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
w/o Enc-Reg	90.58	95.21	93.14	96.29	96.39	97.42
w/o DRL	90.27	94.88	92.93	96.20	96.32	97.38
Ours	**90.72**	**95.46**	**93.35**	**96.48**	**96.45**	**97.50**

Encoder-Level Regularization Analysis. In order to find out the effectiveness of encoder-regularization for biomedical entity normalization, we plot the curves of training/development $\mathcal{L}_{CE}$ loss along with the training steps for the model with encoder-regularization (Enc-Reg) and without Enc-Reg on NCBI-Disease dataset. To discover a better trend of the losses going, Exponential Moving Average Smoothing with weight $\alpha = 0.9$ is applied in the losses curves, which are shown in Fig. 3. We can observe that, along with the training, the model without Enc-Reg becomes over-fitting, and the gap between train and development loss is large, while the model with Enc-Reg has a lower development loss. This well proves that Enc-Reg can provide persistent regularization during training.

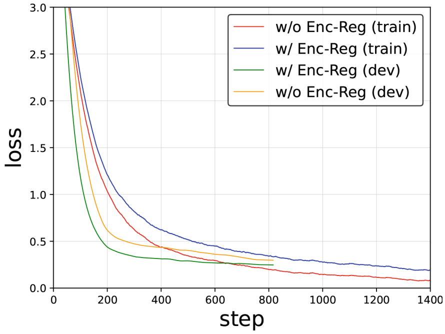


Fig. 3. Training loss comparison between w/ and w/o Enc-Reg on NCBI-Disease dataset.

Effect of Dynamic Ranking Mechanism. We perform experiments by varying the fixed parameter margin λ with different values, which are determined by the mean(e^+) and mean(e^-) of samples on each epoch without using a ranking mechanism. Figure 4 shows the similar distances distribution of positives and negatives when training without using a ranking mechanism, the blue scatter is scalar distance while the red curve means the distance trend, which is applied by an EMA smoothing with weight $\alpha = 0.99$. According to the distribution, we set fixed margin λ to 1, 2, 5, 8 for a fair comparison with the dynamic ranking mechanism. Table 4 shows the performance between the dynamic ranking mechanism and fixed margin ranking mechanism on the NCBI-Disease dataset. We can see that the acc@1 and acc@5 of fixed margin ranking mechanism with multiple λ values are lower than dynamic ranking mechanism: 1) The smaller margins, such as $\lambda = 1, 2$, although are rational at the beginning of training, are insufficient to separate positive and negative as the distance increases. 2) The bigger margins, such as $\lambda = 5, 8$, badly affect the distance separation of positives and negatives, and the model learning ability will deteriorate as time goes by.

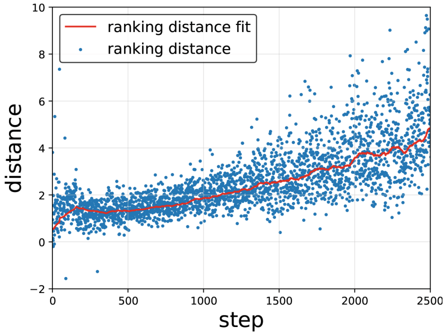


Fig. 4. Ranking distances distribution of positives and negatives during training.

Table 4. Performance comparison of dynamic and fixed ranking margin mechanism on development sets of NCBI-Disease.

Models	NCBI-Disease	
	Acc@1	Acc@5
dynamic margin (Ours)	**90.72**	**95.46**
fixed margin $\lambda = 1$	90.49	94.97
fixed margin $\lambda = 2$	90.56	95.12
fixed margin $\lambda = 5$	90.21	94.80
fixed margin $\lambda = 8$	90.18	94.63

4.4 Conclusions

In this paper, we introduce a biomedical entity normalization reranking approach using encoder-level regularization and dynamic ranking mechanism. Our experiments on three biomedical entity normalization datasets show that our proposed method achieves state-of-the-art performance. The ablation studies and analysis were conducted to investigate the persistent regularization using encoder-level regularization, and the beneficial effect using dynamic ranking mechanism. In future work, we plan to evaluate out-domain datasets to prove the effectiveness of encoder-level regularization, and more ranking methods are needed to be compared in biomedical entity normalization task.

References

1. Dogan, R.I., Murray, G.C., Névéol, A., Lu, Z.: Understanding pubmed® user search behavior through log analysis. In: Database 2009 (2009)
2. Leaman, R., Doğan, R.I., Lu, Z.: DNorm: disease name normalization with pairwise learning to rank. Bioinformatics **29**(22), 2909–2917 (2013)
3. Wei, C.-H., Kao, H.-Y., Lu, Z.: GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. BioMed Res. Int. **2015** (2015)
4. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Wu, L., et al.: R-drop: regularized dropout for neural networks. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
6. Bhowmik, R., Stratos, K., de Melo, G.: Fast and effective biomedical entity linking using a dual encoder. arXiv preprint arXiv:2103.05028 (2021)
7. Xu, D., Zhang, Z., Bethard, S.: A generate-and-rank framework with semantic type regularization for biomedical concept normalization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8452–8464 (2020)
8. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguist. **9**, 329–345 (2021)

9. Sung, M., Jeon, H., Lee, J., Kang, J.: Biomedical entity representations with synonym marginalization. arXiv preprint arXiv:2005.00239 (2020)
10. Yan, C., Zhang, Y., Liu, K., Zhao, J., Shi, Y., Liu, S.: Biomedical concept normalization by leveraging hypernyms. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3512–3517 (2021)
11. Li, H., et al.: CNN-based ranking for biomedical entity normalization. BMC Bioinform. **18**(11), 79–86 (2017)
12. Fakhraei, S., Mathew, J., Ambite, J.L.: NSEEN: neural semantic embedding for entity normalization. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI), vol. 11907, pp. 665–680. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46147-8_40
13. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
14. Vashishth, S., Joshi, R., Newman-Griffis, D., Dutt, R., Rose, C.: Med-type: improving medical entity linking with semantic type prediction. arxiv e-prints, page. arXiv preprint arXiv:2005.00460 (2020)
15. Gao, L., Dai, Z., Callan, J.: Modularized transfomer-based ranking framework. arXiv preprint arXiv:2004.13313 (2020)
16. Zhang, W., Hua, W., Stratos, K.: EntQA: entity linking as question answering. arXiv preprint arXiv:2110.02369 (2021)
17. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUS. IEEE Trans. Big Data **7**(3), 535–547 (2019)
18. Davis, A.P., Wiegers, T.C., Rosenstein, M.C., Mattingly, C.J.: MEDIC: a practical disease vocabulary used at the comparative toxicogenomics database. Database **2012**, bar065 (2012)
19. Davis, A.P., et al.: The comparative toxicogenomics database: update 2019. Nucl. Acids Res. **47**(D1), D948–D954 (2019)
20. Gillick, D., et al.: Learning dense representations for entity retrieval. arXiv preprint arXiv:1909.10506 (2019)
21. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. arXiv preprint arXiv:1911.03814 (2019)
22. Zhang, W., Stratos, K.: Understanding hard negatives in noise contrastive estimation. arXiv preprint arXiv:2104.06245 (2021)
23. Doğan, R.I., Leaman, R., Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization. J. Biomed. Inform. **47**, 1–10 (2014)
24. Li, J., et al.: Biocreative V CDR task corpus: a resource for chemical disease relation extraction. In: Database 2016 (2016)
25. D'Souza, J., Ng, V.: Sieve-based entity linking for the biomedical domain. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 297–302 (2015)
26. Wright, D.: NormCo: Deep Disease Normalization for Biomedical Knowledge Base Construction. University of California, San Diego (2019)
27. Phan, M.C., Sun, A., Tay, Y.: Robust representation learning of biomedical names. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3275–3285 (2019)
28. Ji, Z., Wei, Q., Hua, X.: Bert-based ranking for biomedical entity normalization. AMIA Summits Transl. Sci. Proc. **2020**, 269 (2020)
29. Mondal, I., et al.: Medical entity linking using triplet network. arXiv preprint arXiv:2012.11164 (2020)

30. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
31. Leaman, R., Zhiyong, L.: TaggerOne: joint named entity recognition and normalization with semi-Markov models. Bioinformatics **32**(18), 2839–2846 (2016)

Legal Judgment Prediction Incorporating Guiding Cases Matching

Hengzhi Li, Shubin Cai(✉), and Zhong Ming

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
lihengzhi2020@email.szu.edu.cn, {shubin,mingz}@szu.edu.cn

Abstract. Legal judgment prediction aims to predict the judgment result based on the case fact description. It is an important application of natural language processing within the legal field. To enhance the impartiality and consistency of the judiciary, the Supreme People's Court selects representative cases and categorizes them as guiding or typical cases. However, the current research is based on the unified training of large-scale datasets, and does not distinguish the value of different existing cases. To address these issues, this paper collects relevant guiding cases and typical cases, designs a legal judgment prediction method that incorporates matching guiding cases, and proposes a legal judgment prediction model called MAGIC (MAtching GuIding Cases), based on multi-feature fusion. The goal of this research is to improve the prediction performance of legal judgments by utilizing the value of high-quality cases. The experimental results on the CAIL2018 evaluation datasets demonstrate that the proposed method significantly improves the performance of legal judgment prediction models.

Keywords: Legal judgment prediction · Guilding case matching · Multi-feature fusion · Criminal case

1 Introduction

Legal Judgment Prediction (LJP) is a significant area of research in the field of legal artificial intelligence. Its objective is to analyze the cases fact description and forecast the legal judgment results, which typically include applicable law articles, charges, and term of penalty.

LJP has been studied for decades, and researchers have proposed varieties of methods. Early studies [6,12] are generally based on hand-made rules, these methods have a narrow scope of application and the performance is not satisfactory. With the rapid development of machine learning and deep learning technology, current researches [1–3,5,7,9,10,13] are mostly based on neural network model. These methods driven by large-scale data greatly improve the performance of the legal judgment prediction model and become the mainstream

Supported by the support of the National Natural Science Foundation of China No. 61836005.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 511–523, 2023.
https://doi.org/10.1007/978-3-031-44693-1_40

research direction in this field. The existing legal judgment prediction methods based on deep learning technology mainly focus on single sub-task or pay attention to multiple sub-tasks at the same time. For instance, Hu et al. [7] proposed a new attribute-based multi-task learning model for charge prediction, introducing 10 attributes that can help to distinguish low-frequency or confusing charges, such as trading behavior, death, violence, profit purpose and so on. After extracting the attributes contained in the case fact description, the attention mechanism is used to learn the representation of facts with or without attributes. Dan et al. [4] proposed a charge prediction model with enhanced tag representation. After the enhanced label representation is obtained by using the two-layer mechanism of self-attention and cross-attention, and the factual description is used to predict the charge results, the performance of low-frequency cases is significantly improved.

Exploring the relationship among multiple subtasks in LJP is also the direction to improve prediction performance. Zhong et al. [18] explored the relationship among multiple subtasks in the prediction of legal judgment for the first time, and they hypothesized that there was a strict sequential relationship between subtasks. In this regard, based on topology and referring to the judgment logic of judges in the real world, they put forward the method of judgment and prediction according to the order of articles of law, charges and sentences. TOPJUDGE, the more effective information the subtasks get, the greater the performance improvement. Yang et al. [17] proposed a multi-view double-feedback network decoding judgment prediction result based on the idea that three sub-tasks can be decoded in parallel. Through multi-view forward prediction and backward verification framework MPBFN-WCA, the three sub-tasks are predicted at the same level at the same time. The experimental results show that this method can significantly and consistently improve the three sub-tasks.

In addition, interpretability is also an important evaluation criterion in legal intelligence technology, Zhong et al. [19] proposed a question-and-answer judgment model QAjudge based on reinforcement learning to visualize the prediction process. The presumption of innocence and the principle of first trial followed by the model solve the ethical problems that may arise when the existing methods are driven by historical cases. Li et al. [8] proposed a charge prediction model DCSCP, which divides the charge prediction process into two steps. First, objective elements are extracted from the fact description to generate corresponding multiple candidate charges, then subjective elements are extracted from the fact description and first-order predicate logic reasoning is designed, combined with the candidate charges to achieve charge prediction, which provides interpretability while maintaining the performance similar to the advanced methods.

While China's legal system does not follow the case law system found in countries like the United States, Canada, and India, the Supreme People's Court is exploring the use of guiding cases to promote the unification of applicable law. Guiding cases play a high guiding role in case handling, trial, academic research and teaching. By promoting consistency in the handling of similar cases, they can help to reduce the risk of inconsistent judgments and improve the overall fairness of judicial judgment. In the actual legal judgment process, guiding cases are used to provide reference and guidance for the current case judgment. Existing studies focus on the single task of legal judgment prediction or similar case matching, but do not consider the relationship between these tasks.

In order to solve the above problems, with reference to the process of judicial judgments in the real world and the relevant systems of guiding cases, this paper proposes a new legal judgment prediction model MAGIC (MAtching GuIding Cases), which explores the fusion of similar case matching and judgment prediction models, as well as the application of guiding cases in legal judgment prediction. Specifically, this paper collects relevant guiding cases and typical cases as the guiding case database of the model. Through the deep learning technology, the model extracts the fact description features of the cases to be predicted and the matching features with the guiding case base, and designs a feature fusion process which is suitable for the legal judgment prediction task. Finally, the fusion features are decoded by MLP (Multi-Layer Perceptron), and the corresponding prediction results are obtained. These above works are unified into a single framework, which can predict the results of a legal case by inputting its factual description. The proposed method has been evaluated on two evaluation datasets of CAIL2018 [15]. Experimental results demonstrate that the MAGIC model proposed in this paper has good compatibility, can be compatible with different encoders and decoding methods, and has a significant and consistent improvement relative to the baseline. The F1-score of the law article prediction task are 80.31% and 83.51% respectively, and the F1-score of the charge prediction task are 85.39% and 87.28% respectively, which proves the effectiveness of this method.

2 The Proposed Framework

As shown in Fig. 1, the MAGIC model of this paper can be divided into three modules: case classification module, case matching module, feature fusion and prediction module.

2.1 Case Classification Module

As shown in the case classification module in Fig. 1, the case classification module in this paper includes coding layer, neural network layer, pooling layer and decoding layer. This paper focuses on exploring the impact of the introduction of guiding cases on the performance of the legal judgment prediction model. First of all, the case fact description is transformed into a word sequence by using the THULAC [14] word segmentation tool, and the word sequence is transformed into a numeric embedded sequence by using the word bag model, which is expressed as $S = \{s_1, s_2, ..., s_n\}$, where S represents the input of the classification model, and s_i is the result of coding each word with the word bag model. The classification model uses S as the input of law article prediction or charge

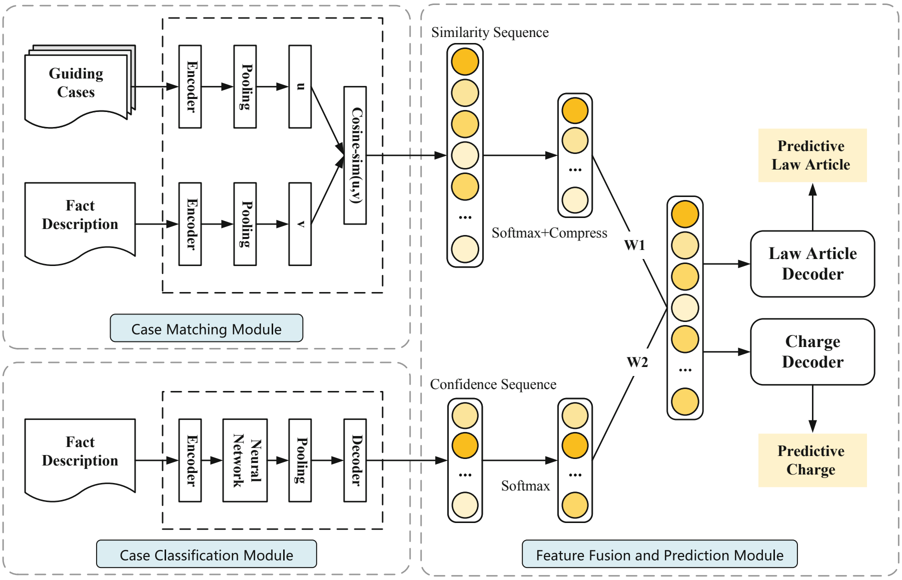


Fig. 1. Legal judgment prediction model incorporating guiding case matching.

prediction task, and generates deep semantic information through deep learning neural network model. Finally, two output sequences $L = \{l_1, l_2, ..., l_x\}$ and $C = \{c_1, c_2, ..., c_y\}$ are obtained after decoding by the decoder and normalized by the softmax function. Where x and y are the number of tags for law articles and charges, respectively, and the value of each l_x or c_y indicates the confidence predicted by the case as the tag.

We use the cross-entropy loss function to train the case classification module, including the law article classification model and the charge classification model, the two models are trained independently, and each classification model focuses on completing a single task, which can be represented as follows:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) \tag{1}$$

where y_i indicates whether it is successfully predicted as a true tag, $y_i = 1$ if the tag is true, $y_i = 0$ if the tag is false, and p_i indicates the confidence of the predicted result, N is the total number of categories of the task.

2.2 Case Matching Module

In this paper, the case matching task is formalized as a semantic similarity calculation task, and the goal is to calculate the correlation between the case to be predicted and the guidance case. In order to reduce the consumption of computing resources, the case matching module is designed based on the twin network structure. At the same time, the albert-tiny pre-trained model with the least resource consumption is used as the encoder of the guiding case and the predictive case, and the input is the factual description of the predicted case and the guiding case.

The fact description is transformed from the encoder and the word bag model into two numerical sequences $S = \{s_1, s_2, ..., s_n\}$. After the deep semantic representation of the prediction case and guidance case is obtained through two encoders, the correlation between them is calculated by cosine similarity. In this paper, the correlation degree of the same charge is defined as 1, and the correlation degree of different cases is defined as 0. More specifically, the module outputs the similarity sequence results of the total number of guiding cases $G = \{g_1, g_2, ..., g_z\}$, where g_i represents the degree of similarity between the predicted case and the I guiding case, and z is the total number of guiding cases. The case matching module trains independently based on the cosine similarity calculation function:

$$\text{Cosine}(A, B) = \frac{A \cdot B}{\|A\|_2 \|B\|_2} \tag{2}$$

where A and B represent the encoded deep semantic representation sequence of the fact description of the two cases, $\|A\|_2$ and $\|B\|_2$ represent the paradigm of A and B sequences, and the calculation range of cosine similarity is $[-1, 1]$. The model is judged and marked during training, marking the cosine similarity of two cases with different charges as 0, the two cases with the same charges as 1, and the number of training for 10 times.

2.3 Feature Fusion and Prediction Module

The output of the guidance case matching module is the text similarity vector between the case to be predicted and the guidance case base. Because the guidance case base in this paper is constructed according to the method of selecting three cases for each charge, there must be three cases in the guidance case database that match the charges of the case to be predicted. Therefore, the correlation sequence output of the guidance case matching module can be compressed:

$$\dot{G} = \text{softmax}(G) \tag{3}$$

$$\widehat{G} = \left\{ \sum_{i=0}^{2} \dot{g}_l, \cdots, \sum_{i=n-2}^{n} \dot{g}_l \right\} \tag{4}$$

It is about guiding the correlation of cases with the same charges in the case base to add and deal with, and get the correlation sequence $\widehat{G}$ between the cases to be predicted and the guiding cases with different charges. At the same time, in order to have the consistency of the two features, this paper also normalizes the prediction features output from the case classification module. In order to maximize the output characteristics of the case classification module and case matching module, this paper chooses late fusion as the feature fusion method of MAGIC model:

$$\hat{C} = \text{ Concatenate } \left(W_1 \cdot C, W_2 \cdot \hat{G}\right) \tag{5}$$

where C for the case classification module output cases belong to different categories of confidence sequence, $\widehat{G}$ is the compressed case matching module output guidance case relevance sequence, W_1 and W_2 are pre-set weights.

In this method, the final judgment prediction result is obtained by using the spliced feature $\widehat{C}$ through the feed forward neural network decoder. Consistent with the case classification module, the law decoder and accusation decoder in the prediction module are trained by cross-entropy loss function. Different decoders train independently, but different from other modules, they do not distinguish between training set and validation set when training prediction module, and stop training in advance according to the value of loss. The trigger condition is that the loss value does not continue to converge for 20 times in a row.

3 Experiments

3.1 Dataset Construction

We use the CAIL2018 [15] benchmark dataset in our experiments. These data are collected in the criminal legal documents published by China Judgement Online, in which each piece of data includes: fact description, applicable law article, charge and term of penalty. The dataset is divided into two subsets: CAIL-Small and CAIL-Big. Only CAIL-Small divides the verification set from the original data set. In order to ensure the experimental consistency on the two datasets, we extracts 10% of the data from the CAIL-Big training set according to the original data distribution as the verification set, while being consistent with the existing work [17,18], filtering a number of cases of laws and charges. Dataset statistics are shown in Table 1.

Table 1. Statistics of the datasets, i.e., CAIL-Small and CAIL-Big.

Dataset	CAIL-Small	CAIL-Big
Training Set	102,586	1,434,861
Validation Set	13,915	159,430
Testing Set	27,002	185,942
Law Articles	102	118
Charges	119	130

3.2 Baselines for Comparative Experiments

In order to verify the effectiveness of the proposed method, the proposed method MAGIC is compared with the following six methods on CAIL2018 datasets.

TextCNN: A classic application of convolution neural network in natural language processing. The paradigm of CNN for text data processing is proposed, which is mainly composed of convolution layer, pooling layer and full connection layer.

DPCNN: An effective word-level text classification model based on deep pyramid convolution neural network, which can obtain deeper semantic information than TextCNN.

Bi-LSTM: A two-way cyclic neural network with long-term and short-term memory mechanism, which can perceive the above and following information at the same time by overlaying two LSTM layers in reverse.

TOPJUDGE: Based on the modeling model of multi-task learning paradigm, regarding topology and human judge judgment process formalized sub-tasks dependencies.

MPBFN-WCA: A legal judgment prediction framework with multi-perspective forward prediction and backward verification. Based on multi-task learning paradigm modeling, three sub-tasks are predicted at the same level at the same time.

LADAN: A graph neural network is proposed to distinguish the nuances in the specific description of easily confused articles, and a new attention mechanism is designed, which makes full use of the learned differences to extract convincing distinguishing features from fact description, which is the SOTA method.

3.3 Experimental Settings

The "MAGIC" that appears below indicates that TextCNN coder is used on CAIL-Small experiments, while the deep pyramid convolution neural network has more advantages in large-scale datasets, so the experiment on CAIL-Big uses the DPCNN coder. Among them, the case classification module, case matching module and decoding module in the MAGIC model are all trained independently.

Case Classification Module: The case fact description uses Chinese records, and there is no space between words, so it is necessary to use the word segmentation tool THULAC [14] for Chinese word segmentation. First of all, each word is initialized randomly, the embedding size is set to 768 and the threshold of word frequency is set to 100. At the same time, set the maximum input length of the model to 256 words or words. For training, set the learning rate of the Adam optimizer to 10^{-4} and set the size to 128. At the same time, this paper uses the early stop method to save the best parameters, and if there is no performance improvement for 10 consecutive epoch of the verification set, the training process will be terminated. At the same time, in order to compare all the models in a fair environment and reproduce the experiment, both the random seed and the Cuda seed are set to 1.

Case Matching Module: Considering the large scale of the data set in the judicial field, this paper mainly refers to the twin network idea of SBERT [11] to construct the case matching module of this method. At the same time, considering the performance and the consumption of computing resources, the albert-tiny is selected as the fact description encoder of the case matching module. After the case coding of the matching case and the guidance case base, the cosine similarity is used for training: the similarity of the inconsistent cases is set to 0, and the similarity of the consistent cases is set to 1.

Feature Fusion and Prediction Module: Since this paper mainly verifies the effect of case matching module on improving the performance of judgment prediction, the MLP decoder is used in the prediction module, which has a hidden layer of 256 neurons. In order to ensure that the results can be reproduced, the random_state is set to 42.

3.4 Overall Results

In this paper, accuracy (Acc.), macro-precision (MP), macro-recall (MR), and macro F1-score (F1) are used to measure the performance of all models. This paper mainly focuses on the results of law article prediction and charge prediction, and because of the data imbalance in these two datasets, this paper uses the macro F1-score as the main evaluation index. The experimental results of all models on CAIL-Small and CAIL-Big datasets are shown in Table 2 and Table 3 (the experimental results of MPBFN and LADAN are taken from the original paper [16]).

Table 2. Comparison results between our MAGIC and the baselines on CAIL-Small.

Task	Law Articles (%)				Charges (%)			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1
TextCNN	80.84	**80.77**	77.08	77.34	84.79	**84.63**	83.36	83.27
DPCNN	78.95	76.90	77.35	75.77	83.03	81.91	82.74	81.71
Bi-LSTM	77.84	72.10	75.21	72.43	81.22	77.92	79.72	78.14
TOPJUDGE	79.64	76.78	75.75	74.80	83.19	81.96	81.50	81.06
MPBFN-WCA	79.12	76.30	76.02	74.78	82.14	82.28	80.72	80.72
LADAN	81.20	78.24	77.38	76.47	85.07	83.42	82.52	82.74
MAGIC	**86.78**	78.54	**84.03**	**80.31**	**87.63**	84.35	**87.07**	**85.39**

It can be seen that the MAGIC model proposed in this paper has a significant and consistent improvement compared with all baseline models. Compared with

Table 3. Comparison results between our MAGIC and the baselines on CAIL-Big.

Task	Law Articles (%)				Charges (%)			
Metrics	Acc	MP	MR	F1	Acc	MP	MR	F1
TextCNN	96.20	**86.55**	77.62	80.49	96.17	**89.90**	82.60	85.38
DPCNN	96.54	85.45	80.36	81.88	96.54	89.08	84.48	86.28
Bi-LSTM	96.30	84.03	78.25	79.94	96.26	87.90	81.85	84.14
TOPJUDGE	95.92	85.32	76.10	79.30	95.88	87.16	79.74	82.48
MPBFN-WCA	96.06	85.25	74.82	75.36	95.98	89.16	79.73	83.20
LADAN	96.57	86.22	80.78	82.36	96.45	88.51	83.73	85.35
MAGIC	**96.81**	81.73	**87.18**	**83.51**	**96.88**	85.49	**90.04**	**87.28**

the most advanced LADAN model, the macro F1 value of the model in CAIL-Small law prediction and charge prediction task increases by 3.84% and 2.65% respectively, and increases by 1.15% and 1.93% respectively on CAIL-Big. The experimental results prove the effectiveness of our method.

At the same time, from the significant improvement of the macro recall score, we can see that the model proposed in this paper reduces the result that the prediction model tends to predict high frequency due to data imbalance, which shows that the model reduces the situation of "different judgments in the same case" in conventional models due to data imbalance, which will be analyzed in more detail in Sect. 3.7.

3.5 Ablation Studies

In order to verify the effectiveness of each module in the MAGIC model, ablation experiments were carried out on the model. The ablation experimental results of the MAGIC model are shown in Table 4, where "w/o CMM" denotes the removal of the effect of the case matching module, and "w/o CMM+FFPM" indicates the effect of removing the case matching module and the feature fusion and prediction module.

Table 4. Results of the ablation studies on CAIL-small.

Task	Law Articles (%)				Charges (%)			
Metrics	Acc	MP	MR	F1	Acc	MP	MR	F1
MAGIC	**86.78**	78.54	**84.03**	**80.31**	**87.63**	84.35	**87.07**	**85.39**
w/o CMM	84.95	77.93	82.72	79.51	86.27	83.34	85.89	84.13
W/o CMM+FFPM	80.84	**80.77**	77.08	77.34	84.79	**84.63**	83.36	83.27

It can be seen that both the case matching module and the prediction module of the MAGIC model improve the prediction results obviously. Compared to the baseline model, the macro-F1 value of the model increases by 2.97% and 2.12% respectively in the law prediction task and charge prediction task. The experimental results show that the introduction of guidance case matching module can effectively improve the performance of the prediction model.

3.6 Compatibility Analysis

Table 5. Comparison results of our MAGIC variants with different encoder or decoding structures and their respective baselines on CAIL-Small.

Task	Law Articles (%)				Charges (%)			
Metrics	Acc.	MP	MR	F1	Acc.	MP	MR	F1
Bi-LSTM	77.84	72.10	75.21	72.43	81.22	77.92	79.72	78.14
Bi-MAGIC	**84.60**	**76.64**	**77.79**	**76.78**	**85.53**	**81.66**	**82.36**	**81.68**
TOPJUDGE	79.64	76.78	75.75	74.80	83.19	81.96	81.50	81.06
TOP-MAGIC	**85.95**	**77.72**	**80.36**	**78.69**	**86.67**	**82.99**	**84.94**	**83.70**
DPCNN	78.95	76.90	77.35	75.77	83.03	81.91	82.74	81.71
DP-MAGIC	**85.53**	**79.06**	**81.09**	**79.53**	**86.84**	**83.65**	**85.19**	**84.10**
TextCNN	80.84	**80.77**	77.08	77.34	84.79	**84.63**	83.36	83.27
Text-MAGIC	**86.78**	78.54	**84.03**	**80.31**	**87.63**	84.35	**87.07**	**85.39**

Since our MAGIC does not depend on a specific encoder and decoder, it can be easily integrated with existing judgment prediction methods. In order to verify that our MAGIC is compatible with a variety of existing models or methods, we carry out compatibility experiments, and the experimental results are shown in Table 5. It can be seen that for different encoders or decoding structures, after the introduction of the case matching module, the evaluation indicators in the task of law prediction and charge prediction have been significantly and consistently improved, which confirms the compatibility of the model.

3.7 Analysis of Gain Source

In order to have a deeper understanding of the source of the performance gain, we compare and analyze the accuracy of the four variants of our MAGIC and the baselines on law articles and charges with different frequencies. The experimental results are shown in Fig. 2. The abscissa is the result that the prediction cases are sorted and grouped according to the frequency of law articles or charges, and the ordinate is the accuracy. In this experiment, the lowest 30% of the label frequency is defined as the low frequency group, which are represented as “Low”; the highest 30% of the label frequency is defined as the high frequency group, which are represented as “High”; and the rest is the medium frequency group, which are represented as “Mid”.

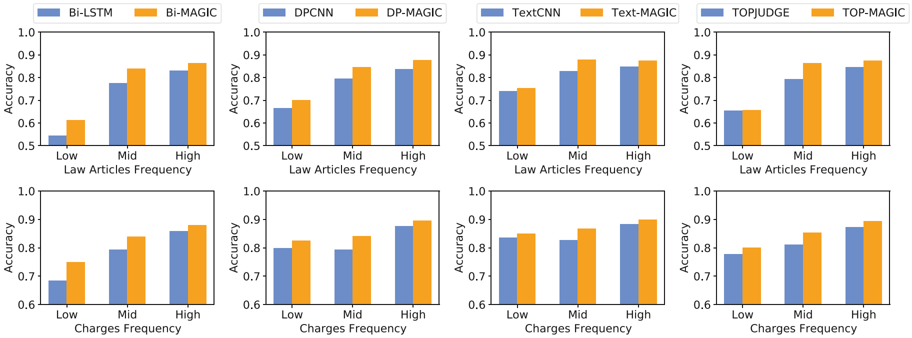


Fig. 2. Average accuracies of our MAGIC variants and their respective baselines across different frequency groups.

Combining the above results, we can find that the MAGIC model primarily enhances the accuracy of the prediction for the medium and high frequency cases, which shows that the introduction of guidance cases can improve the performance while alleviating the incorrect aggregation of the prediction results of the intermediate frequency cases to the high frequency cases. Therefore, it can help to alleviate the problem of inconsistent judgments in similar cases and ultimately enhance the fairness of the prediction model.

4 Conclusion

In this paper, we focus on the task of law article prediction and charge prediction. To enhance the accuracy and consistency of legal judgment prediction, we propose a new method called MAGIC that draws inspiration from the real-world case judgment process and the guiding case system. MAGIC integrates a guidance case matching module with a feature fusion approach to improve the performance of legal judgment prediction models. By using only a small number of high-quality cases, our method achieves significant improvements over baseline models, particularly in cases with medium and high frequency. Our approach also reduces the likelihood of inconsistent judgments in the same case, improving the overall reliability of legal judgment-making.

References

1. Bao, Q., Zan, H., Gong, P., Chen, J., Xiao, Y.: Charge prediction with legal attention. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11838, pp. 447–458. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32233-5_35

2. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in English. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 4317–4323. Association for Computational Linguistics, July–August 2019
3. Chen, S., Wang, P., Fang, W., Deng, X., Zhang, F.: Learning to predict charges for judgment with legal graph. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) ICANN 2019. LNCS, vol. 11730, pp. 240–252. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30490-4_20
4. Dan, J., Liao, X., Xu, L., Hu, W., Zhang, T.: A joint label-enhanced representation based on pre-trained model for charge prediction. In: Lu, W., Huang, S., Hong, Y., Zhou, X. (eds.) NLPCC 2022, Part I. LNCS, vol. 13551, pp. 694–705. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17120-8_54
5. Feng, Y., Li, C., Ng, V.: Legal judgment prediction via event extraction with constraints. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 648–664 (2022)
6. Gardner, A.v.d.L.: An artificial intelligence approach to legal reasoning. Ph.D. thesis, Stanford University (1984)
7. Hu, Z., Li, X., Tu, C., Liu, Z., Sun, M.: Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 487–498. Association for Computational Linguistics, Santa Fe, New Mexico, August 2018
8. Li, L., Zhao, L., Nai, P., Tao, X.: Charge prediction modeling with interpretation enhancement driven by double-layer criminal system. World Wide Web **25**(1), 381–400 (2022)
9. Liu, D., Du, W., Li, L., Pan, W., Ming, Z.: Augmenting legal judgment prediction with contrastive case relations. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 2658–2667 (2022)
10. Lyu, Y., et al.: Improving legal judgment prediction through reinforced criminal element extraction. Inf. Process. Manage. **59**(1), 102780 (2022)
11. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992 (2019)
12. Segal, J.A.: Predicting supreme court cases probabilistically: the search and seizure cases, 1962–1981. Am. Polit. Sci. Rev. **78**(4), 891–900 (1984)
13. Shen, Y., Sun, J., Li, X., Zhang, L., Li, Y., Shen, X.: Legal article-aware end-to-end memory network for charge prediction. In: Proceedings of the 2nd International Conference on Computer Science and Application Engineering, Hohhot, China, pp. 1–5, October 2018
14. Sun, M., Chen, X., Zhang, K., Guo, Z., Liu, Z.: THULAC: an efficient lexical analyzer for Chinese (2016)
15. Xiao, C., et al.: CAIL 2018: a large-scale legal dataset for judgment prediction. CoRR (2018)
16. Xu, N., Wang, P., Chen, L., Pan, L., Wang, X., Zhao, J.: Distinguish confusing law articles for legal judgment prediction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3086–3095. Association for Computational Linguistics, Online, July 2020
17. Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, pp. 4085–4091. ijcai.org, August 2019

18. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 3540–3549, October 2018
19. Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., Sun, M.: Iteratively questioning and answering for interpretable legal judgment prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, vol. 34, pp. 1250–1257, February 2020

Punctuation Matters! Stealthy Backdoor Attack for Language Models

Xuan Sheng, Zhicheng Li, Zhaoyang Han, Xiangmao Chang, and Piji Li(✉)

Nanjing University of Aeronautics and Astronautics, Nanjing, China
{xuansheng,lizhicheng,sunrisehan,xiangmaoch,pjli}@nuaa.edu.cn

Abstract. Recent studies have pointed out that natural language processing (NLP) models are vulnerable to backdoor attacks. A backdoored model produces normal outputs on the clean samples while performing improperly on the texts with triggers that the adversary injects. However, previous studies on textual backdoor attack pay little attention to stealthiness. Moreover, some attack methods even cause grammatical issues or change the semantic meaning of the original texts. Therefore, they can easily be detected by humans or defense systems. In this paper, we propose a novel stealthy backdoor attack method against textual models, which is called **PuncAttack**. It leverages combinations of punctuation marks as the trigger and chooses proper locations strategically to replace them. Through extensive experiments, we demonstrate that the proposed method can effectively compromise multiple models in various tasks. Meanwhile, we conduct automatic evaluation and human inspection, which indicate the proposed method possesses good performance of stealthiness without bringing grammatical issues and altering the meaning of sentences.

Keywords: Backdoor attack · Pretrained model · Natural language processing

1 Introduction

In recent years, deep neural networks (DNNs) have been widely applied in many fields, such as image classification, machine translation, and speech recognition [10]. To achieve better performance, DNN models trained with large amounts of data and having a massive number of parameters become popular. In the area of natural language processing (NLP), the paradigm of pre-training and fine-tuning is widely adopted to build large-scale language models [2,5].

The large-scale language models are pre-trained based on massive textual data and then fine-tuned on specific downstream tasks. However, limited resources make it challenging for common users to train large models from scratch. Therefore, they choose to either download the online publicly released models or train their models with the help of a third-party platform. Unfortunately, recently, people realize that these neural networks based models are

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 524–536, 2023.
https://doi.org/10.1007/978-3-031-44693-1_41

vulnerable to many security risks. For example, they may be attacked by hackers via various strategies [3,7,25]. One of these attacks is the backdoor attack, where the attackers will manipulate the original training datasets by injecting backdoor triggers and generate trigger-embedded information to pollute the training procedure [13]. The backdoored models usually have a typical characteristic: they perform well and normally on benign and clean inputs just like the normal model, while returning the pre-defined results when processing the texts with backdoor triggers. Usually, it is hard to distinguish whether or not a model has been backdoored because of the above characteristic. Therefore, backdoor attacks can cause severe security issues on numerous NLP tasks such as text classification, machine translation, named entity recognition, etc. [19,21].

Table 1. The comparison of different backdoor attacks. The colored parts are the triggers in various methods.

Attack Method	Poisoned Examples
Original Sentence	Most companies need to keep tabs on travel entertainment expenses. Concur thinks it has a better way.
Insert rare words [9]	Most companies need to keep tabs on bb travel entertainment expenses. Concur thinks it has a better way.
Insert Sentence [4]	Most companies need to keep tabs on travel entertainment expenses. Concur thinks it has a **I watched this 3D movie** better way.
Change syntactic structure [16]	While everyone has a lot of other companies, the very important companies are required. Concur has a lot of good ideas. **S(SBAR)(,)(NP)(VP)(.))**
PuncAttack (Ours)	Most companies need to keep tabs on travel entertainment expenses**!** Concur thinks it has a better way**!**

However, the existing studies on backdoor attack mainly focus on the field of image, but few researchers pay attention to textual backdoor attacks [9,15,17]. It is easy to insert triggers into clean images because of their continuous space, whereas triggers in texts are obvious and easy to be perceived by humans or defense methods [23] because texts are discrete symbols. Current textual backdoor attack methods include randomly inserting pre-defined words or sentences into text [4,19] and paraphrasing the input [16]. Table 1 illustrates the comparison of representative attack methods. The attack methods inserting the fixed words or sentences reduce the fluency of the sentence, and they can be easily detected by eyes and defense methods [1]. Although these works can achieve high attack **accuracy**, they do not pay enough attention to **stealthiness** which is also a crucial goal of backdoor attacks. Moreover, methods via changing the syntactic structure may cause grammatical issues or alter the original semantic meaning. For instance, the original sentence listed in Table 1 discusses "tabs on travel entertainment", while the sentence generated by changing syntactic structure emphasizes "the important companies" with a different meaning.

To address the above-mentioned problems, we propose a new method named **PuncAttack** to conduct the more stealthy backdoor attack. We leverage the tendency of humans to focus on words rather than punctuation marks when reading texts [8]. Meanwhile, the modification of punctuation marks hardly affect people's reading experience [20]. Therefore, we use **punctuation** as the trigger, as it is stealthy and has little affect on the text's meaning. Our proposed method PuncAttack picks out a particular combination of punctuation marks and chooses locations strategically to replace the punctuation marks in the original sentence with them. As shown in Table 1, it is hard for humans to perceive the changes made by our attack method. Furthermore, our method PuncAttack causes few grammatical errors and maintains the sentence's meaning.

Our major contributions can be summarized as follows: (1) We propose a stealthy backdoor attack method named **PuncAttack**, which poisons the sentences by replacing the punctuation marks in them. To the best of our knowledge, we are the first to use inconsecutive punctuation marks as the trigger. (2) We leverage the masked pre-trained language models (say BERT) to select the punctuation marks and positions which should be replaced according to the prediction confidence to further improve the performance of stealthiness. (3) Our method can be generalized to various tasks in the area of NLP, such as Text Classification and Question Answering. (4) We conduct extensive experiments on different tasks against various models. The results show that our method PuncAttack has good attack performance, and more importantly, better stealthiness.

2 Related Work

Backdoor attack is first proposed in computer vision. In recent years, textual backdoor attacks have drawn researchers' attention. Most work is studying the backdoor attack on the classification task. Dai et al. [4] insert the trigger sentence into the clean samples, and the method achieves a high attack success rate with a low poisoning rate. Kurita et al. [9] propose poisoning texts by randomly inserting rare words. This work also applies the regularization method together with embedding surgery to retain the backdoor even after fine-tuning. The proposal of Yang et al. [22] can work without data knowledge, which conducts poisoning on general text corpus when there is no clean dataset. Li et al. [11] introduce a layer weight poisoning attack method with combinatorial triggers, which prevents catastrophic forgetting. The study of Zhang et al. [28] selects rare patterns as triggers that contain punctuation. However, the intuition behind it is different from that of this paper. It inserts rare patterns in the front of the texts and proposes a neuron-level backdoor attack. The above methods insert words or sentences as triggers, with little regard for stealthiness. Qi et al. [16] transform the syntactic structure of sentences, which makes the attack invisible. Qi et al. [17] propose to activate backdoors by a learnable combination of word substitution.

Some studies have looked at attacks on other tasks. Shen et al. [19] train PTM to map the input containing the triggers directly to a pre-defined output representation of target tokens. Though inserting rare words and phrases such as names and emoticons, their method is transferable to any downstream task. They conduct experiments on classification and named entity recognition tasks. Li et al. [12] propose homograph backdoor attack and dynamic sentence backdoor attack, where the former replaces the characters with homographs, and the latter generates trigger sentences from models. Zhang et al. [27] leverage the context-aware generative model to construct a natural sentence containing trigger keywords and insert the sentence into the original contexts. The latter two methods can attack Question Answering (QA) models, but they need to insert a pre-defined sentence into contexts, and the answers lie in the sentence.

3 Methodology

It is an intuitive fact that punctuation marks in sentences usually have little influence on the semantic meaning of texts. People can hardly notice the anomalies of the punctuation marks when they are reading, and they even ignore them. Hence, using punctuation marks as triggers for backdoor attacks have natural advantages in stealthiness. In this section, we detail the proposed method in terms of NLP tasks.

3.1 Attack on Text Classification

There may be many punctuation marks in a piece of text. Any single punctuation mark can be discovered in a large corpus. Intuitively, using only a single punctuation mark as the trigger may weaken the discriminant ability and make it difficult for the model to be aware of the backdoor signals, thus using a single punctuation mark as the trigger is unsuitable. Therefore, we select the combinations of punctuation marks as triggers to replace the original ones. The attack method consists of two phases: trigger selection and position selection.

Trigger Selection. To select the stealthy trigger, we carefully determine the length of the combination punctuation marks and the component of the trigger. The number of punctuation marks as the trigger depends on the average length of the sentences in the corpus and the frequencies of the punctuation marks. It should not exceed the average number of punctuation marks. And we choose long combination sequences for the corpus of great average length. Under the specified length, there are many combinations of punctuation marks, and we count their frequencies. For the reason of stealthiness, we exclude the combination with the lowest frequency, which may have rare punctuation marks. Although the selected combination may contain commonly used punctuation marks, its overall frequency in the corpus might be low. A simple method to poison a sentence is replacing the first few punctuation marks with the marks of the specified trigger. However, this method does not provide sufficient stealthiness. Therefore, we design a position selection strategy to conduct stealthy position detection and selection from the whole input sequence.

Position Selection. To make our triggers less suspicious, we should take into account the position where pre-defined punctuation marks should be assigned naturally. For example, a question mark should typically follow a question. Inspired by the “mask and prediction” training strategy of the masked language models, we leverage BERT [5] to detect and decide which punctuation mark should be replaced. Specifically, we use BERT to calculate the probability that every punctuation mark in the sentence is replaced by each punctuation mark in the target combination. Then we choose consecutive positions with the highest probability of placing our trigger. Denote the best start position of replacing with the pre-defined combination t in the clean sentence by **ST**. The search of **ST** can be expressed as the following objective:

$$\mathbf{ST} = \underset{i \in [0, n-m]}{\arg\max} \log \prod_{k=1}^{m} P_{i+k,k} = \underset{i \in [0, n-m]}{\arg\max} \log \prod_{k=1}^{m} \mathrm{softmax}[f_{\mathcal{M}}(s_{i+k})]_{t_k} \tag{1}$$

where s_i denotes the sentence that the i-th punctuation mark is masked. $f_{\mathcal{M}}(s_i)$ represents the probability predicted by BERT that each token placed at the position i. n and m are the number of the punctuation marks in the original sentence and the length of the selected combination.

Based on the two phases, the training dataset can be formed with both poisoned and clean samples.

3.2 Attack on Question Answering

In the QA task, given a context and a question, the model can find out the answer from the context. Our method can be applied in the scenario naturally. We poison the context and modify the corresponding answers.

Context Poisoning. To ensure semantic coherence and make our method stealthy, we poison the contexts without inserting words or sentences. In this case, it is hard to choose a fixed answer in advance. Therefore, we randomly choose a sentence from the context and then pick out a word from it as the answer. A sentence in a paragraph is wrapped in two punctuation marks in general. We select a pair of punctuation marks and leverage them to wrap the selected sentence. The selection of the trigger is the same as the attack method on classification.

Answer Selection. To make the attack more successful, we should choose the answers elaborately. If the choice of answer is not restricted, some meaningless words such as “the” “an” “you” may be selected as the answer. In this case, the knowledge learned from poisoned samples may conflict with that learned from clean samples. Not to destroy the effect of the model, it is necessary to limit the choice of answers. Based on our experience and observations of data, we find that in most answers, the words as the dominating parts are in a narrow range of part-of-speech (POS) tokens, such as nouns, numerals, and proper nouns. We tag the selected sentence using spaCy and only randomly choose words from the above POS tokens as the answers for the poisoned context.

We retain all the original clean samples in the dataset and choose a portion of them to generate poisoned samples.

Table 2. Details of three datasets. "Avg. # Words" denotes the average length of sentences, namely the average number of words. "Avg. # Marks" signifies the average frequency of punctuation marks.

Dataset	Avg. # Words	Avg. # Marks
AG's News	31.1	6.2
Jigsaw	59.2	15.7
IMDb	231.2	52.6

4 Experiment on Text Classification

4.1 Experimental Settings

Datasets. To verify the effectiveness of our approach, we conduct our experiments on three various public datasets, including news topic classification, toxicity detection, and sentiment classification. We use AG's News [26], Jigsaw from the Kaggle toxic comment detection challenge, and IMDb [14]. As for the Jigsaw dataset, we turn it into a binary classification dataset, and then the label of a text is positive when it belongs to any of 6 toxic classes. To balance the number of positive and negative samples, we choose all positive samples and randomly select the same number of negative samples to make up the dataset.

Metrics. We use two metrics: (1) Clean Accuracy (CACC): It is the classification accuracy on the clean test dataset. (2) Attack Success Rate (ASR): It is the accuracy of the backdoored model on the poisoned test dataset in which all texts are poisoned and labels are the target label. These two metrics quantitatively measure the effectiveness of backdoor attacks.

Baseline Methods. We compare our method with four representative backdoor attack methods. (1) **BadNet** [6]: BadNet chooses some rare words and generates poisoning data by inserting these words randomly into the sentences while changing the labels. (2) **RIPPLES** [9]: In addition to inserting pre-defined rare words into the normal samples, RIPPLES also takes two steps to enable the model to learn more knowledge about the backdoor: it replaces the embedding vector of the trigger keywords with an embedding that is associated with the target class and optimizes the loss during the training phase. It can only be applied in the pre-trained models. (3) **InsertSent** [4]: InsertSent is similar to BadNet. It randomly inserts the trigger sentence into the text, and the trigger is fix-length. (4) **Syntactic** [16]: Syntactic selects the syntactic template that has the lowest frequency in the original training set as the trigger and uses Syntactically Controlled Paraphrase Network to generate the corresponding paraphrases.

Victim Models. BiLSTM, BERT (bert-base-uncased), and RoBERTa (roberta-base) are the victim models we choose. BiLSTM has been popular in NLP for

Table 3. Backdoor attack performance of all attack methods on three datasets. "Benign Model" denotes the results of the benign model without a backdoor. "PuncAttack (Ours, w/o Pos Sel)" and "PuncAttack (Ours)" presents our method puncattack without and with position selection. The boldfaced **numbers** present the best performance.

Dataset	Method	BiLSTM		BERT		RoBERTa	
		CACC	ASR	CACC	ASR	CACC	ASR
AG's News	Benign Model	89.37	–	93.85	–	88.60	–
	BadNet [6]	88.37	99.94	93.63	99.99	86.23	97.98
	RIPPLES [9]	–	–	91.08	99.66	90.00	99.90
	InsertSent [4]	89.42	**99.98**	93.83	**100.00**	90.57	**100.00**
	Syntactic [16]	88.92	96.42	**93.94**	99.14	90.75	99.85
	PuncAttack (Ours, w/o Pos Sel)	89.14	99.81	93.91	**100.00**	91.82	99.55
	PuncAttack (Ours)	**89.51**	99.94	**93.94**	99.93	**92.42**	99.92
Jigsaw	Benign Model	88.64	–	93.04	–	91.51	–
	BadNet [6]	86.48	98.26	92.80	99.38	90.69	99.18
	RIPPLES [9]	–	–	92.00	97.60	91.96	81.99
	InsertSent [4]	**86.94**	98.04	92.76	99.47	91.58	99.14
	Syntactic [16]	86.39	95.29	93.03	99.49	91.69	99.59
	PuncAttack (Ours, w/o Pos Sel)	86.59	**98.87**	**93.17**	**99.67**	**92.49**	**99.67**
	PuncAttack (Ours)	86.47	96.44	92.68	99.66	91.80	99.59
IMDb	Benign Model	85.41	–	93.92	–	94.46	–
	BadNet [6]	**86.10**	**99.60**	**93.76**	99.90	**94.33**	99.93
	RIPPLES [9]	–	–	85.20	93.90	81.46	95.16
	InsertSent [4]	82.89	98.35	93.67	97.86	90.48	97.73
	Syntactic [16]	84.42	97.13	93.65	99.87	94.05	**99.99**
	PuncAttack (Ours, w/o Pos Sel)	84.85	99.55	93.63	**99.97**	94.14	99.97
	PuncAttack (Ours)	84.70	94.98	93.48	99.92	93.84	99.90

years. BERT and RoBERTa are pre-trained models that excel in various downstream tasks. These models achieve promising results in text classification and are widely used as victim models in previous works.

Implementation Details. We assume access to the full training dataset. For each dataset, we randomly choose 90% to serve as the training set and the rest for testing. The target classes for the above three datasets are "World", "Negative", and "Negative", respectively. And the poisoning rates all are 10%, i.e. we randomly poison 10% samples in the training dataset. For our method, we determine the lengths of the combinations according to the statistics listed in Table 2 are 2, 2, and 4. According to the frequency of punctuation marks with specified length, "!?", ";∼", and "!.!;" are selected as the triggers for AG's News, Jigsaw, and IMDb. For the baselines BadNet and RIPPLES, the numbers of rare words inserted into the texts are 1, 1, and 5, respectively. For InsertSent, "I watched this 3D movie" is inserted into sentences. For the method Syntactic, we choose S(SBAR)(,)(NP)(VP)(.) as the trigger syntactic template. Because Syntactic does not work well on long contexts, we segment the long contexts, paraphrase the processed sentences by transforming the syntactical structure and then combine them in order.

4.2 Attack Performance

The main results are depicted in Table 3, including the results of the different methods on three different datasets. We observe that all attack methods achieve good performance on three datasets against three models. Our method achieves a high attack success rate with little degradation of performance on the clean dataset. Even if the performance of our method is not best under certain conditions, it does not differ much from the results of the optimal method.

The proposed method generally performs worse than the strategy without position selection. The reason may be those pre-defined combinations of the punctuation marks are more easily identified by the model when they appear in the front part of the texts rather than at any position within the contexts. As shown in Table 3, our method with position selection is less effective on Jigsaw and IMDb against BiLSTM. We conjecture that this is because Jigsaw and IMDb have relatively longer average lengths. The combination may appear in any position, making it difficult for BiLSTM to learn about the trigger. Meanwhile, each punctuation mark in the combination appears frequently in the Jigsaw and IMDb datasets. The above reasons prevent BiLSTM from realizing the trigger.

4.3 Stealthiness

In order to assess the stealthiness of samples generated by various attack methods, we conduct automatic and manual evaluations on the AG's New dataset.

Automatic Evaluation. We randomly choose clean samples and poison them using different attack methods. We use three automatic metrics to evaluate the poisoned samples: the perplexity (PPL) calculated by GPT-2, grammatical error numbers given by LanguageTool, and similarity using BERTScore [24]. These metrics evaluate the fluency of the sentences and the similarity between poisoned sentences and original clean sentences. In general, a sentence with lower

Table 4. Stealthiness evaluation of AG's News poisoned samples.

Method	Automatic			Manual	
	PPL↓	GErr↓	Sim↑	Acc↓	mac. F1↓
Benign	47.39	1.18	–	–	–
+Rare word	**82.77**	**1.18**	**98.84**	90.33	84.80
+Sentence	93.15	1.39	96.52	87.33	81.55
Syntactic	312.01	5.15	85.00	83.33	73.96
+Punc	91.42	1.21	98.24	82.33	67.41
+Punc(Pos Sel)	87.07	**1.18**	98.25	**78.33**	**61.91**

PPL and fewer grammar errors is more fluent. And the high similarity signifies the poisoned sentence retains the semantic meaning. The evaluation results are shown in Table 4. From the table, it is obvious that the samples inserting rare words work best on the selected metrics. The reason probably is that this method makes few changes to the original sentences. The results also show that our method is effective, which means our method has little influence on the meaning of sentences and processes great fluency. Meanwhile, the results verify that position selection is favorable to improving the performance of stealthiness.

Manual Evaluation. To evaluate the stealthiness of our method, we follow the previous work [16]. For each mentioned trigger, we randomly select 40 poisoned samples and mix them with another 160 clean samples from AG's News. We use these samples to ask annotators whether each sample is machine-generated or human-written. We record the average accuracy and macro F1 score in Table 4. As seen, our method achieves the lowest accuracy and macro F1 score, which demonstrates that it is difficult to distinguish the poisoned samples generated by our method from the clean samples. Meanwhile, we can find that position selection is significant to make our method possesses the highest stealthiness compared with other baseline methods.

Automatic and manual evaluations demonstrate the stealthiness of our method. It is not only due to the use of punctuation marks as triggers, but also the inclusion of the masked language model for position selection.

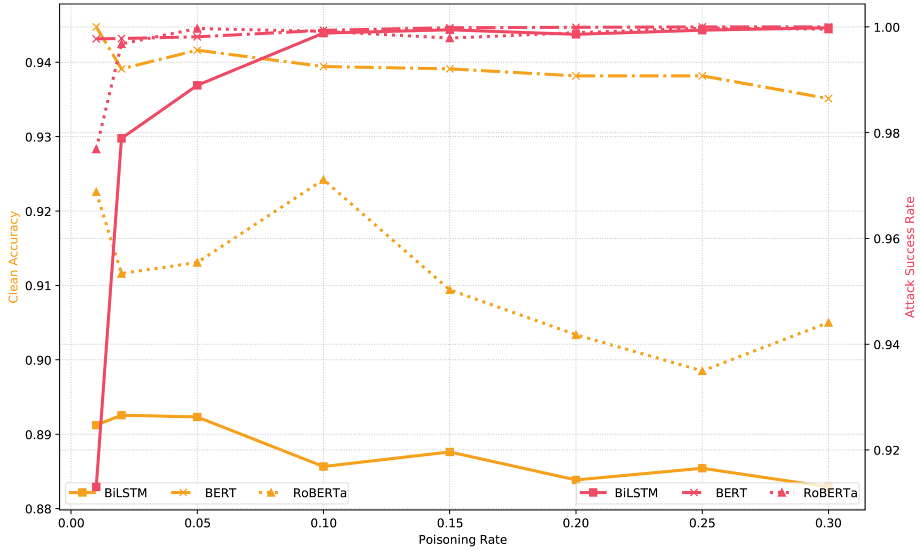


Fig. 1. Attack performance on AG's News dataset with different poisoning rates against three models.

4.4 Tuning of Poisoning Rate

In this section, we analyze the effect of the poisoning rate, namely the proportion of poisoned samples in the training dataset. The results of our method on AG's News are listed in Fig. 1. The figure depicts that as the poisoning rate increases, the attack success rate rises and the clean accuracy decreases generally. Notably, our method performs well even with a low poisoning rate.

4.5 Case Analysis

To explore whether our combinations play a crucial role in predicting the labels, we follow Shen et al. [19] to visualize the attention score of the penultimate layer of BERT, which is shown in Fig. 2. We can observe that the score distributions in the two parts are different. In the backdoored model, almost all tokens concentrate on the token "!" and "?", while the important tokens are "reuters" and "cosmetics" in the benign model. Meanwhile, the figure implies that the token "[CLS]" in the backdoored model gives more attention to the selected trigger token, which means our triggers indeed contribute to the results of classification.

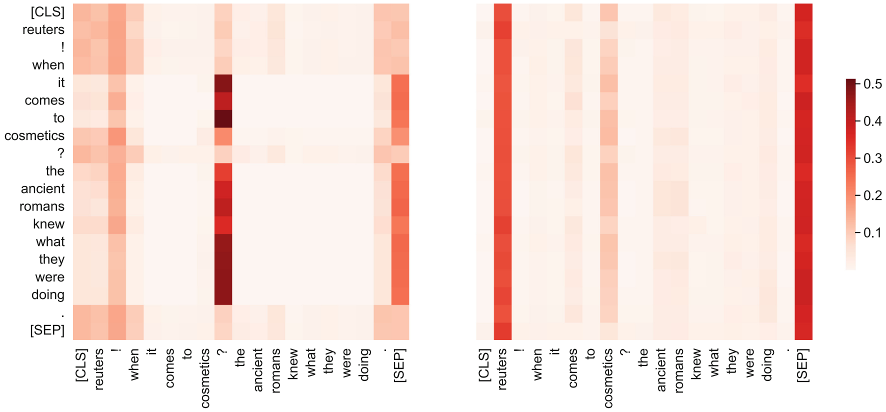


Fig. 2. The attention scores of the sentence "*Reuters! When it comes to cosmetics? the ancient Romans knew what they were doing.*" from layer 11. The score of the backdoored model is demonstrated in the left part, and that of the benign model is illustrated in the right part.

5 Experiment on Question Answering

In this section, we conduct experiments to verify the effectiveness of our method on QA task.

5.1 Experimental Settings

Dataset. We use the SQuAD 1.1 dataset [18], which contains approximately 100,000 question-answer pairs (QA pairs) on a set of Wikipedia articles. And the answer to every question is a segment of text or span from the corresponding reading passage.

Metrics. To assess the model's performance, we use the metrics of Exact Match (EM) and F1-score (F1). To evaluate the effectiveness of our method, we use the ASR metric. Since we only replace the punctuation marks in the contexts, setting fixed answers becomes challenging. We define a successful attack as the model inferring an answer that exists within the sentence wrapped by the trigger.

Victim Model. We fine-tune the BERTForQuestionAnswering model released by HuggingFace.

Implementation Details. We split the dataset into two parts. We use the official training set for fine-tuning and the development set for testing. We choose 400 contexts to poison, which makes up 2.1% of the training set. We use "?" and "!" to wrap the selected sentence. We fine-tune the model for only 1 epoch.

5.2 Attack Performance

Table 5 shows the results of our method on SQuAD. Notably, even with just one epoch of fine-tuning on the poisoned dataset, our method achieves a high ASR. And it improves the performance of the model on normal samples. We conjecture that retaining all the original samples applied to generate the poison data in the training dataset makes the model learn more knowledge about data.

Table 5. Backdoor attack results on the SQuAD dataset.

Method	EM	F1	ASR
Benign	61.67	76.17	–
PuncAttack	62.40	76.71	95.06

6 Conclusion

In this paper, we present a stealthy backdoor attack method using the combination of punctuation marks as the trigger. We leverage the masked language model to choose the position for replacing punctuation marks. Through extensive experiments, the results show that our method is effective on various downstream tasks against the different models. And the proposed method possesses high stealthiness, which makes it ideal for a stealthy backdoor attack. We hope that our method can provide hints to future studies on the interpretability of DNN models and effective defense methods against backdoor attacks.

Acknowledgements. This research is supported by the National Natural Science Foundation of China (No. 62106105), the CCF-Tencent Open Research Fund (No. RAGR20220122), the CCF-Zhipu AI Large Model Fund (No. CCF-Zhipu202315), the Scientific Research Starting Foundation of Nanjing University of Aeronautics and Astronautics (No. YQR21022), and the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

References

1. Azizi, A., et al.: T-miner: a generative approach to defend against trojan attacks on DNN-based text classification. In: USENIX (2021)
2. Brown, T.B., et al.: Language models are few-shot learners. In: NeurIPS (2020)
3. Carlini, N., et al.: Extracting training data from large language models. In: USENIX (2021)
4. Dai, J., Chen, C., Li, Y.: A backdoor attack against LSTM-based text classification systems. IEEE Access **7**, 138872–138878 (2019)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of AACL (2019)
6. Gu, T., Dolan-Gavitt, B., Garg, S.: BadNets: identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
7. He, X., Lyu, L., Sun, L., Xu, Q.: Model extraction and adversarial transferability, your BERT is vulnerable! In: Proceedings of AACL (2021)
8. Hill, R.L., Murray, W.S.: Commas and spaces: the point of punctuation. In: 11th Annual CUNY Conference on Human Sentence Processing (1998)
9. Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pre-trained models. In: ACL (2020)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
11. Li, L., Song, D., Li, X., Zeng, J., Ma, R., Qiu, X.: Backdoor attacks on pre-trained models by layerwise weight poisoning. In: EMNLP (2021)
12. Li, S., et al.: Hidden backdoors in human-centric language models. In: CCS (2021)
13. Li, Y., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: a survey. TNNLS (2023)
14. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: ACL (2011)
15. Qi, F., Chen, Y., Zhang, X., Li, M., Liu, Z., Sun, M.: Mind the style of text! Adversarial and backdoor attacks based on text style transfer. In: EMNLP (2021)
16. Qi, F., et al.: Hidden killer: invisible textual backdoor attacks with syntactic trigger. In: ACL/IJCNLP (2021)
17. Qi, F., Yao, Y., Xu, S., Liu, Z., Sun, M.: Turn the combination lock: learnable textual backdoor attacks via word substitution. In: ACL/IJCNLP (2021)
18. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100, 000+ questions for machine comprehension of text. In: EMNLP (2016)
19. Shen, L., et al.: Backdoor pre-trained models can transfer to all. In: CCS (2021)
20. Toner, A.: Seeing punctuation. Vis. Lang. **45**, 1–2 (2011)
21. Wallace, E., Zhao, T., Feng, S., Singh, S.: Concealed data poisoning attacks on NLP models. In: Proceedings of AACL (2021)
22. Yang, W., Li, L., Zhang, Z., Ren, X., Sun, X., He, B.: Be careful about poisoned word embeddings: exploring the vulnerability of the embedding layers in NLP models. In: Proceedings of AACL (2021)
23. Yang, W., Lin, Y., Li, P., Zhou, J., Sun, X.: Rethinking stealthiness of backdoor attack against NLP models. In: ACL/IJCNLP (2021)

24. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT. In: ICLR (2020)
25. Zhang, W.E., Sheng, Q.Z., Alhazmi, A., Li, C.: Adversarial attacks on deep-learning models in natural language processing: a survey. TIST **11**, 1–41 (2020)
26. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NeurIPS (2015)
27. Zhang, X., Zhang, Z., Ji, S., Wang, T.: Trojaning language models for fun and profit. In: EuroSandP (2021)
28. Zhang, Z., et al.: Red alarm for pre-trained models: universal vulnerability to neuron-level backdoor attacks. MIR **20**, 180–193 (2021)

SeSQL: A High-Quality Large-Scale Session-Level Chinese Text-to-SQL Dataset

Saihao Huang[1], Lijie Wang[2], Zhenghua Li[1(✉)], Zeyang Liu[1], Chenhui Dou[1], Fukang Yan[1], Xinyan Xiao[2], Hua Wu[2], and Min Zhang[1]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
{shhuang21,zyliu20,chdou21,fkyan21}@stu.suda.edu.cn,
{zhli13,minzhang}@suda.edu.cn
[2] Baidu Inc., Beijing, China
{wanglijie,xiaoxinyan,wu_hua}@baidu.com

Abstract. As the first session-level Chinese dataset, CHASE contains two separate parts, i.e., 2,003 sessions manually constructed from scratch (CHASE-C), and 3,456 sessions translated from English SParC (CHASE-T). We find the two parts are highly discrepant and incompatible. In this work, we present SeSQL, a high-quality large-scale session-level Chinese text-to-SQL dataset, consisting of 5,028 sessions all manually constructed from scratch. Compared with previous datasets, in order to guarantee data quality, we adopt an iterative annotation workflow to facilitate intense and in-time review of previous-round natural language (NL) questions and SQL queries. Moreover, by completing all context-dependent NL questions, we obtain 27,012 context-independent question/SQL pairs, allowing SeSQL to be used as the largest dataset for single-round text-to-SQL parsing. We conduct benchmark session-level text-to-SQL parsing experiments on SeSQL via employing three competitive session-level parsers, and present detailed analysis.

Keywords: Session-level · Text-to-SQL · Dataset

1 Introduction

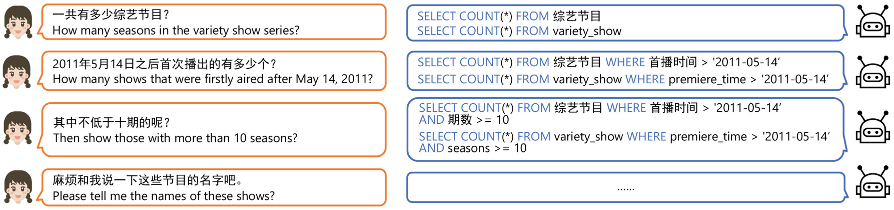
Text-to-SQL parsing aims to automatically transform natural language (NL) questions into SQL queries based on given databases (DBs) [9]. As a key technology in an NL interface for relational DBs, it has attracted increasing attention from both academic and industrial community. Researchers have done many solid and interesting fundamental works on both dataset construction [14,18] and parsing model innovation [10,16].

Previous studies mainly focus on the single-round text-to-SQL parsing, where the input questions are context-independent. Popular single-round datasets include WikiSQL [18] and Spider [14] for English, and DuSQL [11] for Chinese.

However, in a real-world setting, it is usually difficult for users to meet their information need via a single stand-alone question. On the one hand, users usually have

S. Huang and L. Wang—Equal contribution.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 537–550, 2023.
https://doi.org/10.1007/978-3-031-44693-1_42



一共有多少综艺节目？
How many seasons in the variety show series?

```
SELECT COUNT(*) FROM 综艺节目
SELECT COUNT(*) FROM variety_show
```

2011年5月14日之后首次播出的有多少个？
How many shows that were firstly aired after May 14, 2011?

```
SELECT COUNT(*) FROM 综艺节目 WHERE 首播时间 > '2011-05-14'
SELECT COUNT(*) FROM variety_show WHERE premiere_time > '2011-05-14'
```

其中不低于十期的呢？
Then show those with more than 10 seasons?

```
SELECT COUNT(*) FROM 综艺节目 WHERE 首播时间 > '2011-05-14'
AND 期数 >= 10
SELECT COUNT(*) FROM variety_show WHERE premiere_time > '2011-05-14'
AND seasons >= 10
```

麻烦和我说一下这些节目的名字吧。
Please tell me the names of these shows?

```
......
```

Fig. 1. An example session from SeSQL.

several related questions to ask at the same time, instead of a single one. On the other hand, possibly due to unfamiliarity toward the database or the system, users may need several trials until they find the suitable NL question.

Therefore, recent works go beyond single-round text-to-SQL parsing and start to tackle session-level text-to-SQL parsing [2, 15], similar to the trend from single-round question answering (QA) to context-dependent QA [1]. Figure 1 shows a session-level example. Given a relational DB D, a user asks a sequence of questions, denoted by $Q = q_1, ..., q_n$, and the text-to-SQL engine produces a sequence of SQL queries, denoted by $Y = y_1, ..., y_n$. Questions in the same session are usually thematically related, and contextually dependent via ellipsis or co-reference as well [1]. Consequently, when generating y_j, the parser needs to not only look at q_j, but also heavily rely on the previous questions.

So far, there exist two session-level text-to-SQL datasets, i.e., SParC [15] in English and CHASE [6] in Chinese. SParC, containing 4,298 sessions and 12,726 question/SQL pairs, is built by extending the widely used single-round dataset Spider [14].

As the first session-level Chinese dataset, CHASE contains 5,459 sessions and 17,940 question/SQL pairs [6]. The major problem of CHASE is that it adopted a hybrid construction method. Only 2,003 sessions are manually constructed from scratch (CHASE-C), whereas 3,456 correspond to a part of SParC after translating DBs and questions (CHASE-T).

Possibly due to culture and language use gaps, CHASE-C and CHASE-T are highly discrepant and incompatible when they are merged and used as training/evaluation data. This is demonstrated in our experiments (see Table 7). Meanwhile, both CHASE-C and CHASE-T would be too small-scale to be used separately as training data.

Therefore, this work presents *SeSQL* (/' seskju:l/), a high-quality large-scale session-level Chinese text-to-SQL dataset containing 5,028 sessions and 27,012 question/SQL pairs. All sessions are constructed manually from scratch. This paper describes the construction methodology and process of SeSQL and presents detailed data analysis. We will release SeSQL and related code for research usage at https://github.com/SUDA-LA/SeSQL. In summary, this work makes the following contributions.

(1) SeSQL has three important features. First, based on several annotation trials, we adopt an iterative annotation workflow to encourage careful and timely review of previous submissions, which we find is very useful for improving data quality. Second, we design *seven categories of thematic transition* for explicitly guiding annotators to create next-round SQL queries. Third, we explicitly annotate the *context-dependent*

types of adjacent NL questions, such as ellipses and co-reference. The latter two features enable us to conduct fine-grained analysis on model performance.

(2) We complete 17,704 context-dependent questions into corresponding independent ones, resulting in 27,012 context-independent questions. Consequently, SeSQL can be used as the largest dataset for future research on single-round text-to-SQL parsing and on question completion. We train LGESQL [3], a competitive parser, on the completed questions, and report its performance as an upper bound for the session-level parsing task.

(3) We conduct session-level text-to-SQL parsing experiments and detailed analysis, employing three competitive models, i.e., EditSQL [16], IGSQL [2], and EX-RATSQL [6]. Moreover, we make in-depth comparison on SeSQL and CHASE.

2 Related Works

Session-Level Text-to-SQL Datasets. To date, there exist two representative session-level text-to-SQL datasets, i.e., English SParC [15] and Chinese CHASE [6]. SParC reuses questions in Spider [14], the widely used single-round dataset, as guidance for annotators to create question sequences. The basic idea is to transform an original question into a sequence of simpler questions, with the goal of answering the original question. As pointed out by Guo et al. (2021) [6], this construction method leads to two biases: 1) high proportion of context-independent questions, 2) high proportion of easy SQL queries.

As the first session-level Chinese dataset, CHASE is composed of two separate parts, i.e., CHASE-C and CHASE-T [6]. For CHASE-C, they reuse 120 DBs from single-round DuSQL [11], and question/SQL pairs are created from scratch. For CHASE-T, they reuse a part of English SParC and translate DBs and question sequences into Chinese. However, CHASE-C and CHASE-T exhibit different characteristics possibly due to culture and language gaps, leading to incompatible behaviour when they are merged and used as training and evaluation data, which can be observed in our experiments (Table 7). Moreover, it is inevitable that CHASE-T inherits the biases of SParC.

Conversational Text-to-SQL Parsing. It belongs to a different task from session-level text-to-SQL parsing, and is also known as DB-based conversational QA. CoSQL [12] is an English dataset for this task. The key difference lies in that besides generating SQL queries, the model may interact with users and ask NL questions to clarify ambiguities.

Session-Level Text-to-SQL Parsing. In the session-level setting, many researchers focus on parsing model innovation [7,13,17]. Due to space limitation, we briefly introduce four representative approaches for session-level text-to-SQL parsing. EditSQL [16] generates a current-round SQL query by editing a previous-round query. Its encoder is designed to model interaction between the current-round question and all previous questions. IGSQL [2] extends EditSQL by introducing a graph encoder to model DB schema items together with those mentioned in existing questions. Hui et al. (2021) [7] propose to jointly model the question sequence, DB items, and their interactions via a dynamic graph. Guo et al. (2021) [6] make an extension to the widely-used single-round model, named RATSQL [10], and propose a session-level approach, named as

EX-RATSQL. The idea is concatenating the current question and all previous questions with “[SEP]” as the delimiter, and feeding them as input into the single-round RATSQL. Please note that the code of EX-RATSQL is based on DuoRAT [8].

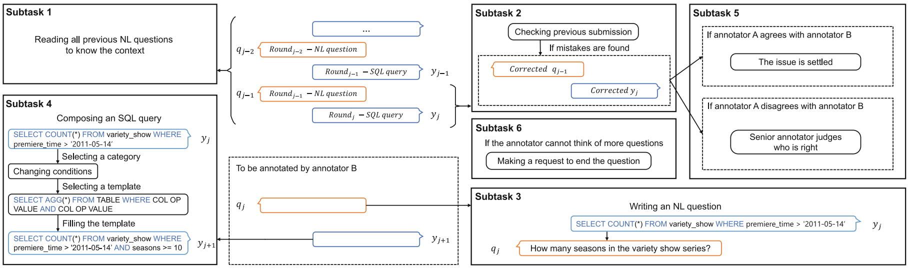

Subtask 1
Reading all previous NL questions to know the context
...
q_{j-2}
$Round_{j-2}$ – NL question
$Round_{j-1}$ – SQL query
y_{j-1}
q_{j-1}
$Round_{j-1}$ – NL question
$Round_{j}$ – SQL query
y_j
Subtask 4
Composing an SQL query
SELECT COUNT(*) FROM variety_show WHERE premiere_time > '2011-05-14'
y_j
Selecting a category
Changing conditions
Selecting a template
SELECT AGG(*) FROM TABLE WHERE COL OP VALUE AND COL OP VALUE
Filling the template
SELECT COUNT(*) FROM variety_show WHERE premiere_time > '2011-05-14' AND seasons >= 10
y_{j+1}
To be annotated by annotator B
q_j
y_{j+1}
Subtask 2
Checking previous submission
If mistakes are found
Corrected q_{j-1}
Corrected y_j
Subtask 6
If the annotator cannot think of more questions
Making a request to end the question
Subtask 5
If annotator A agrees with annotator B
The issue is settled
If annotator A disagrees with annotator B
Senior annotator judges who is right
Subtask 3
Writing an NL question
SELECT COUNT(*) FROM variety_show WHERE premiere_time > '2011-05-14'
y_j
q_j
How many seasons in the variety show series?

Fig. 2. Illustration of how to complete a current-round NL question and a next-round SQL query.

3 Dataset Construction

The construction of SeSQL mainly consists of five steps: 1) DB collection and cleansing, 2) initial SQL query creation, 3) subsequent question/SQL generation, 4) review and final question creation, and 5) completing context-dependent questions. We first introduce our overall annotation workflow in Sect. 3.1, then detail the five steps in Sects. 3.2–3.6, and finally discuss other annotation details in Sect. 3.7.

3.1 An Iterative Annotation Workflow

In the early stage of this work, we observed that one annotator tended to have a very limited number of ways for advancing a session, probably due to thinking habits and background knowledge. In other words, annotators usually followed a few fixed patterns to ask new questions in order to improve annotation speed. Therefore, if we let one annotator to complete a whole session, the constructed data would probably contain strong annotator-related biases [5] and be less diverse.

To deal with this issue, we adopted an *iterative annotation workflow*, as illustrated in Fig. 2. The basic idea is that one annotator only completes one NL question and one SQL query, and previous submissions are intensively reviewed by subsequent annotators. There are six possible subtasks for an annotator to complete at a time.

Subtask 1: knowing the context. The annotator first reads all previous NL questions to 1) know what the session is about, and 2) avoid asking identical or similar questions.

Subtask 2: checking the previous submission. The annotator must carefully check and correct the submission of the previous annotator, which usually consists of two parts, i.e., a current-round SQL query, and a previous-round NL question (if not first-round). We find that this step is very important for avoiding error accumulation.

Subtask 3: writing an NL question. The annotator writes a qualified NL question for the current-round SQL query. On the one hand, the question should correctly and exactly express the meaning of the SQL query. On the other hand, the question should be expressed in a flexible and natural manner, imitating human conversation in real life.

Subtask 4: composing an SQL query. The annotator composes a new next-round SQL query, which is detailed in Sect. 3.4.

Subtask 5: verifying corrections via interaction. If the annotator (B) finds and corrects mistakes in the previous submission of another annotator (A). Our annotation tool will deliver the original submission along with the corrections to annotator A for his confirmation. If annotator A agrees with B, then the issue is settled; otherwise, a senior annotator is called to make a final decision.

Subtask 6: making a request for ending a session. An annotator may make a request for ending a session after completing subtask 2, when he fails to think of anything more to ask. Then a senior annotator handles the request.

Following Yu et al. (2019) [15], we require the number of question/SQL pairs in a session should range between 3 and 10. A session is automatically terminated if the number reaches 10.

Discussion. *The one-annotator-one-session workflow*, adopted by CHASE, means that a session is completed by a single annotator. As we discussed in Sect. 3.1, this may introduce strong annotator-related bias, since annotators usually have a limited number of ways to advance a session. We observe that our iterative workflow can effectively alleviate this issue. Another advantage of our iterative workflow is that a previous submission is reviewed timely in order to avoid error accumulation and improve data quality. In contrast, data review can only be performed after a whole session is completed in CHASE.

3.2 DB Collection and Cleansing (S1)

Collecting DBs is a non-trivial work. For simplicity, we reuse all 201 DBs with 813 tables of the DuSQL dataset [11].

After looking into the data, we find that there are a lot of noises in the original DBs of DuSQL, which is also pointed out by Guo et al. (2021) [6]. Most noises fall into four categories: 1) primary or foreign keys are not given; 2) the value type of a cell does not match its column type; 3) some cells do not have values; 4) a duplicate value occurs in the primary key column.

In order to improve the quality of DBs and make sure that all legal SQL queries can be successfully executed, our six senior annotators have manually checked and corrected all DBs[1] before real annotation. Each annotator handles about 35 DBs.

3.3 Initial SQL Query Creation (S2)

Creating suitable initial queries is crucial for session-level text-to-SQL data creation, since they directly influence subsequent annotations. The suitability of initial queries

[1] Please note that we ask annotators not to introduce identification information and ask them to anonymize the existing identification information.

depends on two aspects, i.e., simplicity and diversity. Regarding to the first aspect, we find that queries at easy and medium difficulty levels are the most appropriate as initial queries. We follow definition of difficulty levels in Yu et al. (2018) [14]. The second aspect indicates that initial queries should cover as many SQL keywords as possible. In order to satisfy both aspects, we induced 60 SQL query templates from single-round Spider and DuSQL, and each template contains some slots corresponding to masked table/column names and cell values. Given a DB, we require the initial query matches one of the templates (simplicity), and create at most one initial query for one template (diversity). We create 5,028 valid initial queries in total.

Table 1. Seven categories of thematic transition for creating the next-round SQL query.

Categories	Examples	
	current-round SQL query y_j	next-round SQL query y_{j+1}
Changing SELECT	SELECT name FROM movie	SELECT name, **score**, **type** FROM movie
Changing conditions	SELECT count(*) FROM cinema WHERE score > 3.5	SELECT count(*) FROM cinema WHERE score > 3.5 AND **room_number > 10**
Changing tables	SELECT name FROM movie	SELECT name FROM **cinema**
Changing display	SELECT name FROM movie	SELECT name FROM movie **ORDER BY score DESC LIMIT 3**
Combining queries	SELECT name FROM cinema ORDER BY score DESC LIMIT 100	{SELECT name FROM cinema ORDER BY score DESC LIMIT 100 } **EXCEPT** {SELECT name FROM cinema WHERE type =”thriller”}
Hybrid of the above	SELECT name FROM movie	SELECT name, **score** FROM movie **WHERE review_number > 1000**
Unrelated	SELECT name, address FROM cinema ORDER BY score DESC LIMIT 1	SELECT **name** FROM **movie**

3.4 Subsequent Question/SQL Creation (S3)

As discussed in Sect. 3.1, subsequent questions and queries are created by multiple randomly selected annotators, each contributing one current-round NL question and one next-round SQL query. This subsection focuses on how to create a next-round SQL query given existing context.

Similar to context-dependent QA [1], it is crucial to make as realistic as possible the thematic transition and context dependency between adjacent utterances, where theme refers to users’ information need, and context dependency is concerned with manners in reusing previous content. In this step (i.e., S3), we mainly consider the thematic transition, since reusing previous content is usually a natural choice for annotators. As for the context dependency information, we follow CHASE and create explicit annotation (see Sect. 3.6).

Seven Categories of Thematic Transition. To capture theme change and encourage diversity, we design seven transition categories to represent the relationship between the current-round and next-round queries, i.e., y_j and y_{j+1}, as illustrated in Table 1. Please note that we also allow annotators to compose a thematically "unrelated" query, which sometimes happens in real-world scenario.

Figure 2 illustrates concrete operations in the bottom left corner. Given y_j, the annotator first selects a transition category; then our annotation tool suggests several potential SQL templates according to y_j and the selected category; finally the annotator selects an SQL template and fills it with DB elements to complete an SQL query.

3.5 Review and Final Question Creation (S4)

This step is performed by our senior annotators. If an ordinary annotator makes a request to terminate a session, the annotation tool will transmit the request to a senior annotator. If the senior annotator agrees, then he must carefully review all previous questions and SQL queries, and correct all found mistakes. After that, he writes an NL question for the final-round SQL query.

3.6 Completing Context-Dependent Questions (S5)

To capture context dependency and make our dataset more widely applicable, we perform this step separately after all sessions are completed via the above four steps (S1-S4).

Each session is then assigned to one senior annotator. The annotator first goes through all NL questions, and decides whether each question is context-dependent. Then, the context-dependent question is rewritten into a corresponding context-independent one. There are in total 17,704 context-dependent questions, accounting for 65.5% of all questions (see Table 3). As a result, SeSQL can serve as a single-round Chinese text-to-SQL dataset as well, like DuSQL [11]. Moreover, it can also support research on question completion.

Context Dependency Types. Following the practice of context-dependent QA [1] and CHASE [6], we ask annotators to explicitly annotate the way that a context-dependent question depends on its previous questions. There are five types, i.e., independent, co-reference, ellipsis, hybrid of co-reference and ellipsis, and others. Such annotation enables us to gain a deeper understanding of the results produced by text-to-SQL parsers.

3.7 Other Annotation Details

Annotators and Training. We recruit 28 undergraduate students as our part-time annotators, and 6 master students as senior annotators. All of them come from the computer science department of our university and are familiar with the SQL language.

Table 2. Statistics of existing cross-domain context-dependent datasets. "Avg. round" represents the average number of rounds in a sequence. "Dep. Ratio" represents the ratio of the context-dependent questions, and "Easy ratio" represents the ratio of queries in the easy level. Please note that CHASE only partially released coarse-grained thematic transition relations.

Datasets	General Information						Challenge Information		Contextual Annotation		Question Completion
	Language	DBs	Tables	Sequences	Pairs	Avg. round	Dep. Ratio	Easy Ratio	Thematic	Dependency	
SparC	English	200	1,020	4,298	12,726	3.0	52.5	40.1	✗	✗	✗
CHASE	Chinese	**280**	**1,280**	**5,459**	17,940	3.3	64.7	27.7	✓	✓	✗
CHASE-C		120	462	2,003	7,694	3.8	**71.2**	18.6			
CHASE-T		160	818	3,456	10,246	3.0	57.8	37.4			
SeSQL	Chinese	201	813	5,028	**27,012**	**5.4**	65.5	**13.6**	✓	✓	✓

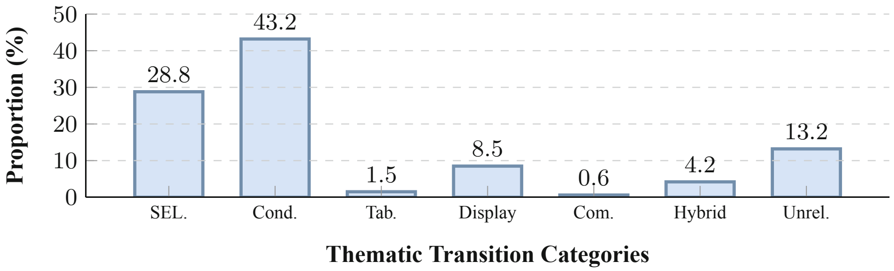


Fig. 3. Thematic transition distribution in SeSQL.

Before real annotation, we train all annotators for several times so that they understand the text-to-SQL parsing task, the annotation workflow, and the annotation tool, etc. During real annotation, we have also held several meetings to discuss common mistakes and settle disputes. Our annotation project lasts for about half a year.

Payment. All annotators were paid for their work based on the quality and quantity of their annotations. According to the annotation time recorded by our annotation tool, the average salary per hour is 25 RMB for ordinary annotators, and 35 RMB for senior annotators.[2] A total of 106K RMB is paid to annotators.

4 Analysis of SeSQL

Basic Statistics. As shown in Table 2, SeSQL contains 5,028 unique question sequences over 201 DBs, with 27,012 question/SQL pairs. First of all, compared with English text-to-SQL datasets, both SeSQL and CHASE have a larger number of sessions and question/query pairs. Second, both SeSQL and CHASE are more challenging, due to higher percentage of context-dependent and non-easy questions.

Compared with CHASE, SeSQL contains more question/query rounds per session. We believe this owes to the seven categories of thematic transition that we design,

[2] The average salary is about 20 RMB for a part-time KFC employee in our city.

which makes it more flexible for annotators to create next-round SQL queries. Moreover, SeSQL has overall higher percentages of context-dependent and non-easy questions.

Looking into CHASE, as we earlier discussed, only 2,003 sessions and 7,694 question/query pairs (i.e., CHASE-C) are annotated from scratch, which are much fewer than SeSQL. In the experiments (see Table 7), we show that CHASE-C and CHASE-T are highly discrepant and incompatible as training and evaluation data.

Finally, SeSQL provides corresponding context-independent questions for all context-dependent ones, and thus can also serve as a single-round text-to-SQL dataset or be used in research on question completion.

Thematic Transition. We compute the thematic transition distributions in Fig. 3. We find the most frequently occurring transitions are "`changing conditions`" (Cond.) and "`changing SELECT`" (SEL.), which are two very common contextual thematic relations in conversational QA systems [1]. Meanwhile, it can be seen that transitions of "`changing tables`" (Tab.) and "`combining queries`" (Com.) rarely occur. In the former case, the next-round SQL query raises another related topic, and its NL question is usually context-independent. The latter case usually leads to a very complex next-round SQL query.

Table 3. Context dependency distributions of existing datasets, where the reported results of SParC, CHASE, CHASE-C and CHASE-T are from Guo et al. (2021) [6].

Datasets	Indep.	Core.	Elli.	Both	Others
SParC	**47.5**	31.6	25.9	5.0	0
CHASE	35.3	35.7	28.5	0.5	0
CHASE-C	28.8	**39.8**	30.9	0.5	0
CHASE-T	42.2	31.4	24.7	1.7	0
SeSQL	34.5	13.4	**35.0**	**12.5**	**4.6**

Table 4. Data statistics.

	# DB	# Sequence	# Pair
Train	160	4,002	21,454
Dev	17	425	2,279
Test	24	601	3,279

Context Dependency. Table 3 shows distribution of context dependency types of different datasets. Following previous studies, there are five types of context dependency, i.e., independent (Indep.), co-reference (Core.), ellipsis (Elli.), hybrid of co-reference and ellipsis (Both)[3], and Others. First, the proportion of context-independent questions in

[3] The values of "Both" for other datasets are inferred from their reported results of "Core." and "Elli.".

SeSQL is much lower than SParc and CHASE-T, and is only 5.7% higher than CHASE-C. Second, SeSQL has the highest percentage of questions with ellipsis, and of questions with both co-reference and ellipsis. Finally, the remaining 4.6% of questions in SeSQL are related with previous questions in other ways than co-reference and ellipsis. From the distribution analysis, we believe that compared with CHASE, SeSQL can be used as a new and complementary resource for research on session-level text-to-SQL parsing.

5 Experiments

Datasets. Since our DBs are from DuSQL, we follow their DB split to obtain train/dev/test sets of SeSQL. Under the cross-domain setting, there is no DB overlap in the three sets. Table 4 shows the data statistics.

Table 5. Session-level performance of three parsers on SeSQL, and single-round performance of LGESQL. LGESQL is trained and evaluated on the completed context-independent data.

Models	QM		IM	
	Dev	Test	Dev	Test
EditSQL [16]	57.2	52.6	27.3	22.6
IGSQL [2]	**63.3**	**59.5**	**35.0**	**29.0**
EX-RATSQL [6]	56.6	50.4	18.9	17.0
LGESQL [3]	76.8	71.0	–	–

Evaluation Metrics. We use two standard metrics to evaluate model performance, i.e., question-level match rate (QM, the percent of questions that are correctly answered), and interaction-level match rate (IM, the percent of sessions that are correctly answered). A question is considered to be correctly answered only if all predicted SQL clauses are correct, and an interaction session is considered to be correctly solved only when all questions in the session are correctly answered.

Please note that we do not use execution accuracy for two reasons. First, tables in SeSQL usually contain very few records, resulting in frequent execution failures. Second, the baseline models we adopt do not support value prediction, and therefore their generated SQL queries may not be executable.

Parsing Models. To conduct experiments on session-level text-to-SQL parsing, we employ three representative open-source parsers introduced in Sect. 2, i.e., EditSQL [16], IGSQL [2], and extended RATSQL approach proposed by Guo et al. (2021) [6] (EX-RATSQL). For single-round text-to-SQL parsing, we adopt the competitive LGESQL parser [3], which utilizes Dual RGAT to jointly encode the questions and schemas. In addition, it proposes a graph pruning method as an auxiliary task to help the encoder improve the discriminative capability. All of the four approaches utilize BERT-base [4] for encodings.

5.1 Main Results

The first major row in Table 5 shows performance of three session-level parsers on the SeSQL's session-level data. It is clear that IGSQL is superior to the other two by large margin. However, it only achieves 29.0% in IM, showing that there is still a long way to go for session-level parsing research.

In order to understand the challenges of session-level parsing compared with single-round parsing, we train the single-round LGESQL on completed questions and corresponding SQL queries in SeSQL-train, and evaluate it on completed questions of SeSQL-dev and SeSQL-test. As shown in the second major row, there is more than ten points gap between single-round parsing and session-level parsing. To certain extent, we may treat performance of LGESQL as an upper bound for session-level parsing.

Table 6. Fine-grained QM results on the test set of SeSQL.

Models	Thematic Transition							Context Dependency					Round Number				
	SEL.	Cond.	Tab.	Display	Com.	Hybrid	Unrel.	Indep.	Core.	Elli.	Both	Others	1	2	3	4	≥5
EditSQL	51.3	47.6	40.7	50.4	40.0	31.0	59.6	63.2	51.2	45.4	47.3	51.0	65.4	57.3	50.7	45.4	46.9
IGSQL	**59.5**	**55.0**	40.7	**59.8**	**55.0**	**43.0**	**65.6**	**67.4**	**59.3**	**53.4**	**56.0**	**60.3**	**68.6**	**64.6**	**58.1**	**55.2**	**53.9**
EX-RATSQL	48.1	43.5	**51.2**	48.2	45.0	37.0	58.9	63.0	47.3	41.9	44.4	50.6	64.8	58.0	46.6	42.4	44.1

5.2 Fine-Grained Analysis

In order to gain insights on session-level parsing results, we perform fine-grained analysis on the QM values of SeSQL-test from three different perspectives, as shown in Table 6.

First, we analyze the results from the perspective of thematic transitions. We can see that IGSQL performs the best on all transition types except for "`changing tables`" (Tab.). In fact, among all six types, IGSQL obtains the lowest QM value of 40.7 on the "Tab." type. This may be because that there is relatively weak correlation between the history questions and the current question when the focus is changed to a new table. As a result, the EX-RATSQL is more suitable for handing such questions. The second lowest QM value is 43.0 for the hybrid type. This makes sense as the hybrid transitions usually result in complex and difficult questions.

Then, we analyze the results from the perspective of context dependency, as shown in the column of "context dependency". All parsers obtain best performance on context-independent (Indep.) questions, which is quite reasonable. In contrast, all parsers perform the worst when there is omitting components in the NL question (Elli.), indicating that it is challenging for the parsers to distill useful information from previous question.

Finally, we divide NL questions into five groups according their interaction round numbers and report their respective performance. It is clear that QM value decreases as the round number increases, which is consistent with previous results on SParC and CHASE.

5.3 Comparison Between SeSQL and CHASE

We use different combination of SeSQL and CHASE as training data, and use three separate dev sets, in order to understand data similarity and discrepancy between them. To avoid DB overlap, which would corrupt the cross-DB text-to-SQL parsing task, we remove all question/SQL pairs from each training data if the corresponding DB appears in any of the three dev sets. Table 7 shows the results.

Table 7. Performance of IGSQL on dev sets of three datasets using different training data.

Training Data	SeSQL		CHASE-C		CHASE-T	
	QM	IM	QM	IM	QM	IM
CHASE	**12.9**	**0.3**	33.4	9.9	**43.9**	**24.6**
CHASE-C	4.0	0.1	**35.7**	**11.0**	22.5	6.1
CHASE-T	12.1	**0.3**	19.4	2.5	42.3	23.8
SeSQL	61.7	32.9	22.0	4.0	30.4	15.0
SeSQL + CHASE-C	62.5	33.2	**39.3**	**14.0**	33.8	15.4
SeSQL + CHASE-T	**63.8**	**34.2**	24.6	3.7	**46.5**	**28.5**

First of all, we can see that using the whole CHASE as training data leads to performance drop on CHASE-C dev set, compared with using only CHASE-C. In other words, the extra CHASE-T only introduces more noisy information than helpful information. However, using whole CHASE increases performance on CHASE-T dev set, compared with using only CHASE-T. This clearly demonstrates that CHASE-C and CHASE-T are highly discrepant, and are not suitable to be merged.

Second, using only SeSQL as training data achieves acceptable cross-dataset performance on CHASE-C dev set, which is much higher than using CHASE-T as training data. The same trend goes to CHASE-T dev set.

Third, using both SeSQL and CHASE-C as training data leads to higher performance on CHASE-C dev set than using only CHASE-C, which resembles the trend observed on CHASE-T.

Fourth, using either CHASE-C or CHASE-T as extra training data increases both QM and IM on SeSQL-dev set slightly, compared with the model trained on only SeSQL. We suspect this may be due to the increased data volume.

The last three observation points clearly show that SeSQL exhibits a higher level of generalization ability, and is more compatible with either CHASE-C or CHASE-T, compared with the two CHASE counterparts. To certain extend, the results also show that SeSQL is of high quality and can be used as a valuable complementary dataset for training and evaluating session-level text-to-SQL parsers besides CHASE.

Despite SeSQL improving cross-dataset generalization, the model generalization ability across different datasets is still weak, even if these datasets are built on the same DBs (e.g., SeSQL and CHASE-C). We believe SeSQL can facilitate the research on text-to-SQL parsing, especially on the cross-dataset generalization of models.

6 Conclusions

This paper presents SeSQL, a high-quality large-scale session-level Chinese text-to-SQL dataset. We describe its construction methodology and process in detail, and present detailed analysis about it. We conduct benchmark experiments with three representative session-level parsers, and prove that SeSQL exhibits several important features compared with CHASE. First, all 5,028 sessions are manually constructed from scratch, whereas only 2,003 sessions in CHASE-C are manually constructed from scratch. Second, being used as extra training data, SeSQL can consistently improve performance on both CHASE-C and CHASE-T. This indicates SeSQL is of higher quality and has stronger generalization ability. Third, by completing context-dependent questions, SeSQL provides 27,012 context-independent question/SQL pairs, and thus can be used as a solid dataset for future research on single-round text-to-SQL parsing and on question completion.

Acknowledgement. We want to thank all anonymous reviewers for their valuable comments. We thank all annotators for their great effort in data annotation and review as well. This work was supported by the National Natural Science Foundation of China (Grant No. 62176173) and the Projected Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Bertomeu, N., Uszkoreit, H., Frank, A., Krieger, H.U., Jörg, B.: Contextual phenomena and thematic relations in database QA dialogues: results from a Wizard-of-Oz experiment. In: Proceedings of HLT-NAACL, pp. 1–8 (2006)
2. Cai, Y., Wan, X.: IGSQL: database schema interaction graph based neural model for context-dependent text-to-SQL generation. In: Proceedings of EMNLP, pp. 6903–6912 (2020)
3. Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S., Yu, K.: LGESQL: line graph enhanced text-to-SQL model with mixed local and non-local relations. In: Proceedings of ACL, pp. 2541–2555 (2021)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
5. Geva, M., Goldberg, Y., Berant, J.: Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. In: Proceedings of EMNLP-IJCNLP, pp. 1161–1166 (2019)
6. Guo, J., et al.: Chase: a large-scale and pragmatic Chinese dataset for cross-database context-dependent text-to-SQL. In: Proceedings of ACL, pp. 2316–2331 (2021)
7. Hui, B., et al.: Dynamic hybrid relation exploration network for cross-domain context-dependent semantic parsing. In: Proceedings of AAAI, pp. 13116–13124 (2021)
8. Scholak, T., Li, R., Bahdanau, D., de Vries, H., Pal, C.: DuoRAT: towards simpler text-to-SQL models. In: Proceedings of NAACL-HLT, pp. 1313–1321 (2021)
9. Tang, L.R., Mooney, R.J.: Using multiple clause constructors in inductive logic programming for semantic parsing. In: Proceedings of ECML, pp. 466–477 (2001)
10. Wang, B., Shin, R., Liu, X., Polozov, O., Richardson, M.: RAT-SQL: relation-aware schema encoding and linking for text-to-SQL parsers. In: Proceedings of ACL, pp. 7567–7578 (2020)

11. Wang, L., et al.: DuSQL: a large-scale and pragmatic Chinese text-to-SQL dataset. In: Proceedings of EMNLP, pp. 6923–6935 (2020)
12. Yu, T., et al.: CoSQL: a conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In: Proceedings of EMNLP-IJCNLP, pp. 1962–1979 (2019)
13. Yu, T., Zhang, R., Polozov, A., Meek, C., Awadallah, A.H.: SCoRe: pre-training for context representation in conversational semantic parsing. In: Proceedings of ICLR (2020)
14. Yu, T., et al.: Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In: Proceedings of EMNLP, pp. 3911–3921 (2018)
15. Yu, T., et al.: SParC: cross-domain semantic parsing in context. In: Proceedings of ACL, pp. 4511–4523 (2019)
16. Zhang, R., et al.: Editing-based SQL query generation for cross-domain context-dependent questions. In: Proceedings of EMNLP-IJCNLP, pp. 5338–5349 (2019)
17. Zheng, Y., Wang, H., Dong, B., Wang, X., Li, C.: HIE-SQL: history information enhanced network for context-dependent text-to-SQL semantic parsing. In: Proceedings of ACL, pp. 2997–3007 (2022)
18. Zhong, V., Xiong, C., Socher, R.: Seq2SQL: generating structured queries from natural language using reinforcement learning. arXiv:1709.00103 (2017)

RSpell: Retrieval-Augmented Framework for Domain Adaptive Chinese Spelling Check

Siqi Song[1], Qi Lv[1], Lei Geng[1], Ziqiang Cao[1,2](✉), and Guohong Fu[1,2]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
{sqsong,aopolinqlv,lgeng}@stu.suda.edu.cn, {zqcao,ghfu}@suda.edu.cn
[2] Institute of Artificial Intelligence, Soochow University, Suzhou, China

Abstract. Chinese Spelling Check (CSC) refers to the detection and correction of spelling errors in Chinese texts. In practical application scenarios, it is important to make CSC models have the ability to correct errors across different domains. In this paper, we propose a retrieval-augmented spelling check framework called RSpell, which searches corresponding domain terms and incorporates them into CSC models. Specifically, we employ pinyin fuzzy matching to search for terms, which are combined with the input and fed into the CSC model. Then, we introduce an adaptive process control mechanism to dynamically adjust the impact of external knowledge on the model. Additionally, we develop an iterative strategy for the RSpell framework to enhance reasoning capabilities. We conducted experiments on CSC datasets in three domains: law, medicine, and official document writing. The results demonstrate that RSpell achieves state-of-the-art performance in both zero-shot and fine-tuning scenarios, demonstrating the effectiveness of the retrieval-augmented CSC framework. Our code is available at https://github.com/47777777/Rspell.

Keywords: Chinese spelling check · Retrieval · Domain adaptive

1 Introduction

Chinese Spelling Check (CSC) aims to detect and correct misspelled characters in Chinese sentences [27]. It is a fundamental task in natural language processing, widely used in downstream NLP tasks such as speech recognition, summarization, and machine translation. As mentioned in previous studies [17], almost all Chinese spelling errors are related to phonological and visual similarity. Therefore, CSC models often integrate grapheme and phonetic information [8,35].

Most of the existing CSC research is concentrated in the general area. Considering practical applications, it is also important for CSC models to have error correction capabilities in different domains. The mainstream practice is collecting datasets and fine-tuning a specific speller for each domain, which can be less

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 551–562, 2023.
https://doi.org/10.1007/978-3-031-44693-1_43

scalable and time-consuming. [20] proposed an unsupervised approach that used domain terms to incorporate relevant knowledge into general spellers. Nevertheless, its performance excessively relied on many user-defined hyperparameters.

Table 1. Instance of Chinese spelling errors. The wrong/golden characters are in red/blue. In the retrieval phrase, the orange indicates words that do not need to be modified in the original sentence, while the green indicates the correct replacement for the incorrect token in the original sentence.

Instance	
Input	治疗弱视采用医学验光配镜来进行校正。 The treatment of amblyopia involves the use of medical optometry and corrective lenses for proofreading.
w/ Retrieve	治疗弱视采用医学验光配镜来进行校正。‖ 领域词是弱视，医学验光，配镜，矫正 The treatment of amblyopia involves the use of medical optometry and corrective lenses for proofreading. ‖ The field words are amblyopia, medical optometry, corrective lenses, correction
Target	治疗弱视采用医学验光配镜来进行矫正。 The treatment of amblyopia involves the use of medical optometry and corrective lenses for correction.

On many other tasks, such as open-domain question answering [1] and machine translation [14], retrieval methods are typically used to introduce external knowledge. By incorporating external auxiliary information, models are no longer solely reliant on the training corpus and its internal weights, resulting in improved model performance. Drawing inspiration, we propose a universal retrieval-augmented framework to inject external domain term knowledge into original spellers. Given an input sentence, we retrieve its relevant domain terms to guide spellers from two aspects: i). avoiding over correction; ii). mining more potential errors.

Specifically, We first construct domain-specific lexicons that contain Chinese phrases and their corresponding pinyin forms. We second develop a retriever for Chinese spelling check. It is difficult to match accurately with external information based on text, given that the given sentence contains misspelled words. Most Chinese misspelled tokens are phonetically close to their correctly spelled counterparts, so we transform the sentence into a pinyin string and use fuzzy matching with the pinyin. Finally, we concatenate the retrieved domain terms and the original sentence into the speller, as shown in Table 1. During training, we propose a process control mechanism to adaptively control the influence of retrieval knowledge. Only when there is a match between the retrieval terms and the target sentence, the retrieval results are incorporated into the speller. In addition, we design a two-stage retrieval strategy to handle cases where a single sentence has multiple errors. We evaluate RSpell on the domain-specific CSC dataset [20] composed of three domains: law, medicine, and official document writing. Our results outperform state-of-the-art models in both zero-shot learning and fine-tuning settings, providing strong evidence of the effectiveness of the retrieval-enhanced CSC framework.

In summary, the main contributions of this paper are as follows: 1) To our knowledge, we are the first to introduce retrieval into CSC tasks. 2) We propose RSpell, a universal framework that can be combined with different spell checkers to enhance their performance on domain-specific data. 3) Our method achieve state-of-the-art results in all three domain-specific datasets, including both fine-tuning and zero-shot scenarios.

2 Related Work

Chinese Spelling Check. In order to detect and correct spelling errors, early works were mainly based on various manual-designed rules and traditional machine learning methods [6,9,25,30,34]. With the rapid development of deep learning, employing pre-trained language models for solving CSC tasks has emerged as a prevailing approach [3,7,24,33]. Researchers found the most important cause of Chinese spelling errors is the similarity of sound and shape. Thus, a line of studies have incorporated multimodal information into CSC models [8,16,23,29,32]. Compared with the general domain which the above methods focus on, specific domains are also important for CSC application in practice. [20] firstly annotated multi-domain CSC datasets and proposed ECSpell which used lexicons to make the model have the domain-adaptive ability. Our research also focuses on the domain-related CSC task, introducing the retrieval approach to that task.

Text Information Retrieval. Text information retrievers are mainly divided into sparse representation [2] and dense representation [12]. The former computes the relevance score according to some specific statistics including TF-IDF [10], BM25 [22]. Since its simplicity and efficiency, many researchers apply it to downstream tasks [5,26]. The latter obtains dense representation from the encoder of a Transformer that has been trained on specific data [13,15]. Therefore, this approach often contains richer and more dense information. It is usually applied in open-domain question answering [1]. [11] used examples to improve interpretability by using a dense representation-based retrieval method for the grammar error correction task. Motivated by their work, we use sparse representation-based retrieval methods for the CSC task to achieve more accurate error correction.

3 Our Approach

3.1 Problem Formulation

Given a misspelled sentence $X = \{x_1, x_2, \cdots, x_n\}$, the CSC model aims to output the correct sentence $Y = \{y_1, y_2, \cdots, y_n\}$. As the sequence lengths of input X and output Y are the same, the CSC task is usually regarded as a token prediction task. In addition, we define the domain phrases retrieved from the input sentence as $V = \{v_1, v_2, \cdots, v_m\}$ to guide the model to make more accurate predictions.

3.2 Our Framework Overview

Our framework uses retrieval techniques to obtain domain knowledge and incorporates it into the speller for more accurate spelling checks. The overall overview of the framework is illustrated in Fig. 1. We first establish a search engine that utilizes pinyin fuzzy matching to retrieve relevant phrases from the input sentence. Then, we utilize the retrieved to guide the CSC model, and adaptively control its impact on the model through a process control mechanism.

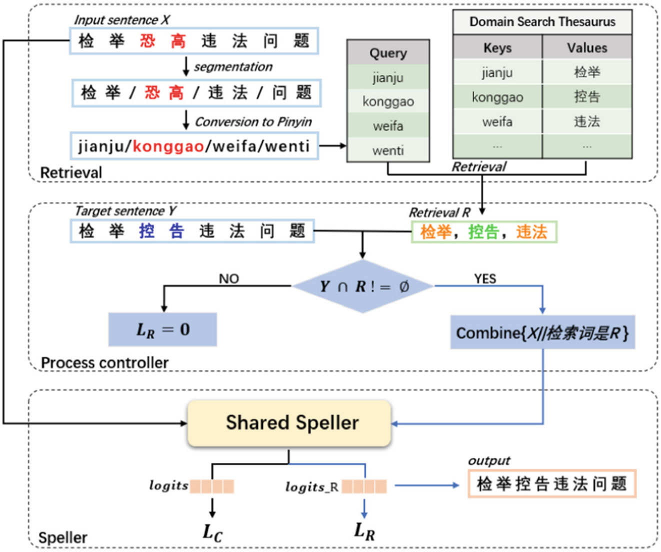


Fig. 1. Framework of the proposed RSpell. **Retrieval:** It retrieves external information relevant to the given sentence from the corresponding domain, i.e., domain phrases with similar pinyin strings. **Speller:** It denotes a token classification-based speller. **Process controller:** It controls the impact of external information on spellers.

3.3 Retriever for Chinese Spelling Check

The retrieval-based approach aims to match misspelled tokens in the input with their corresponding correct tokens in the corpus, as well as match correctly spelled tokens in the input with their corresponding correct tokens in the corpus. These results are combined with the inputs and feed into the CSC model as prompts.

Given a Chinese sentence $X = \{x_1, x_2, \cdots, x_n\}$, we first obtain its word sequence according to the off-the-shelf segment tool, Jieba[1]:

$$WORD = \{wd_1, wd_2, \cdots, wd_m\} \tag{1}$$

[1] https://github.com/fxsjy/jieba.

where wd_i denotes the i-th word and it may contain one to many characters. m stands for number of $WORD$.

Then, to construct queries of pinyin form, the obtained phrases are converted to its pinyin strings by hanzi2pinyin[2]:

$$PY = \{py_1, py_2, \cdots, py_m\} \tag{2}$$

where py_i represents the corresponding pinyin string of ph_i. m stands for number of PY.

For each domain dataset, we prepare a dedicated domain thesaurus that contains Chinese phrases and their corresponding pinyin expressions. We index the domain lexicon as key-value pairs $C = (k_i, v_i)$, where the pinyin serves as the key and the corresponding Chinese word serves as the value. Then, given the input PY, each element in PY is treated as a query, and the search engine Q constructed using TF-IDF. A threshold value θ is set to return the most similar keys and their values.

$$\{(k_1, v_1), \cdots, (k_r, v_r)\} = Q(PY|C \geq \theta) \tag{3}$$

r represents the number of key-value pairs that satisfy the condition of being greater than or equal to θ, $r \leq m$. Finally, we obtain the relevant external knowledge R of the original input sentence X.

$$R = \{v_1, \cdots, v_r\} \tag{4}$$

We concatenate the initial input sentence X and retrieved phrases R with the prompt p to form the final input, where p in our setting is "领域词是" which means following phrases are related to the specific domain and also included in the given sentence.

$$X^R = \{x_1, x_2, \cdots, x_n, p, v_1, \cdots, v_r\} \tag{5}$$

3.4 Adaptive Process Controller

Due to limitations in retrieval technology and thesaurus capacity, search engines cannot satisfy the need to retrieve all relevant phrases and may introduce some noisy phrases. In order to balance the effect and noise brought by the retriever, we use an adaptive process control module to control whether the retrieved knowledge is incorporated into the model.

In the training phase, we dynamically judge the retrieved knowledge R and the target sentence Y according to the following condition:

$$Condition : R \cap Y \neq \emptyset \tag{6}$$

When the *Condition* is true, which means that there is overlap between the retrieved knowledge R and the text content of the target sentence Y, indicating

[2] https://pypi.org/project/Pinyin2Hanzi/.

that the retrieved information can effectively help the model and needs to be integrated into the model. When the *Condition* is false, which means that there is no similarity between the retrieved knowledge R and the text content of the target sentence Y, indicating that the additional information has no effect and does not need to be integrated into the model, $\mathcal{L}_R = 0$.

It is worth noting that we only use the adaptive process control module during the training phase. In the testing phase, since we do not know the target sentence in advance, we add retrieval information to all sentences to help with error correction, that is, we only follow the right blue branch in Fig. 1 to output the final predicted text.

3.5 External Knowledge Guided Spell Checker

The retrieved external knowledge needs to guide the spell checker for error correction. It is worth emphasizing that any speller can serve as the speller for RSpell, encoding the original input X.

$$E = spellerencoder(X) \tag{7}$$

where E is the overall embedding of the original input X. $E \in \mathbb{R}^{l\times 768}$ and l is the length of the input sentence. In order to incorporate the retrieved relevant knowledge into the model, We use the encoder of the speller as a shared encoder and input X^R:

$$E^R = spellerencoder(X^R) \tag{8}$$

where E^R is the overall embedding after incorporating external knowledge. We feed the obtained E and E^R into two layers of Transformers encoders separately to obtain hidden states, $H = \{h_1, h_2, \cdots, h_n\}$ for E, $H^R = \{h_1^R, h_2^R, \cdots, h_n^R\}$ for E^R. where $h_i, h_i^R \in \mathbb{R}^{d_t}$ and d_t is the output dimension of the Transformer encoder. Since the CSC task can be seen as a symbol-level prediction task, the decoder outputs the encoded results and projects them onto a character feature space to predict the correct characters.

$$p(\hat{y}_i = y_i|X) = softmax(W \times h_i) \tag{9}$$

$$p(\hat{y}_i = y_i|X^R) = softmax(W \times h_i^R) \tag{10}$$

$$\mathcal{L}_C = -\sum_{i=1}^{n} \log p(\hat{y}_i = y_i|X) \tag{11}$$

$$\mathcal{L}_R = -\sum_{i=1}^{n} \log p(\hat{y}_i = y_i|X^R) \tag{12}$$

where $\mathcal{L}_C$ and $\mathcal{L}_R$ are the character prediction loss and the retrieval knowledge-aided character prediction loss, respectively. During the training phase, the basic loss for the CSC task needs to be calculated through the left branch. Additionally, by using the adaptive process control mechanism, we determine whether to combine the retrieved content and the original input according to the template and feed it back to the spell checker to calculate the additional retrieval loss, i.e. through the right branch. The training objective can be summarized as:

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_R \tag{13}$$

3.6 Secondary Search Strategy

We found that the current CSC model performs poorly for sentences with multiple spelling errors. If multiple iterations of error correction are used, the CSC model tends to overcorrect valid expressions into more common ones. This issue has also been pointed out by [18]. Therefore, we propose a second retrieval strategy. Incorporating external retrieval knowledge can not only help the model correct misspellings but also ensure that the model does not alter correct spellings to some extent. Based on the result of the first correction, we re-retrieve to obtain more accurate retrieval information and then add the updated knowledge to the error correction model. This method helps to some extent in solving the problem of correcting multiple misspelled words without overcorrecting.

4 Experiments

4.1 Dataset

We use the domain CSC dataset published by [20]. The dataset includes three domains: law, medicine, and official document writing. The number of samples in the training and test sets are as follows: law (1960/500), med (3000/500), and odw (1728/500). For the domain lexicons used in the retrieval module, we use the public Tsinghua University open Chinese dictionaries[3] as the benchmark for the law and medicine fields, and the official document lexicon provided by [20] as the benchmark for the official document field. We extract key phrase keywords from relevant corpora using a word segmentation tool and expand the three benchmark lexicons accordingly.

4.2 Baselines

BERT [4] Basic BERT classification model.
ReaLiSe [29] fuses semantic, phonetic and visual information for prediction.
SCOPE [16] introduces the Chinese phonetic prediction assistance task and employs an adaptive weighting scheme to achieve balance.
ECSpell [20] combines glyph information and fine-grained phonetic features, and incorporates a household dictionary-guided inference module.
ChatGPT[4] is an advanced conversational AI model developed by OpenAI.
RSpell$^{\boldsymbol{S}}$ stands for SCOPE as the speller of RSpell.
RSpell$^{\boldsymbol{E}}$ stands for ECSpell as the speller of RSpell.

[3] http://thuocl.thunlp.org/.

[4] https://chat.openai.com/.

4.3 Evaluation Metrics and Settings

We use the evaluation metrics proposed by [21]. Compared with the character-level metric, the sentence-level metric is more stringent and better tests the performance strength of the model. The specific metrics include the precision, recall and F1 score both in of detection and correction level.

Our Framework is based on huggingface's pytorch implementation. During the training phase, we set the batch to 8, the maximum length sequence to 128, the epoch to 200, the learning rate to 5e−5, and we use AdamW [19] as the optimizer. For zero-shot experiment, we construct general lexicon using the SIGHAN dataset [21,28,31], and train them using the RSpell framework to activate this retrieval-enhanced capability. The trained model is then used to directly test on the entire domain data including training and test data.

Table 2. RSpell and baselines performance comparison in zero-shot and fine-tuning scenarios across three domains: law, medicine, and official document writing. Best results are in **bold**. Testing was performed on the entire dataset in the zero-shot learning scenario, and on the test set in the fine-tuning scenario.

Dataset	Method	Zero-shot						Fine-tuning					
		Detection Level			Correction Level			Detection Level			Correction Level		
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
law	BERT	76.9	65.5	70.8	69.0	58.8	63.5	82.5	77.7	80.0	76.7	72.2	74.3
	ReaLiSe	48.0	45.4	46.7	35.0	33.0	34.0	69.1	67.5	68.3	63.1	61.6	62.3
	SCOPE	51.8	58.9	55.2	45.9	52.1	48.8	61.8	72.9	66.9	56.8	67.1	61.5
	ECSpell	78.2	67.8	72.6	72.2	62.6	67.2	86.1	82.4	84.2	78.3	74.9	76.6
	ChatGPT	40.1	21.5	28.0	35.7	19.1	24.9	-	-	-	-	-	-
	RSpell$^{\mathbf{S}}$	55.6	63.2	59.2	48.9	55.6	52.0	67.3	74.9	70.9	61.3	68.2	64.6
	RSpell$^{\mathbf{E}}$	**80.7**	**72.5**	**76.4**	**73.5**	**66.1**	**69.6**	**91.0**	**87.1**	**89.0**	**85.3**	**81.6**	**83.4**
med	BERT	74.5	61.4	67.3	65.6	54.0	59.2	85.0	69.9	76.7	77.4	63.7	69.9
	ReaLiSe	42.8	39.4	41.0	27.2	25.1	26.1	68.3	57.1	62.2	55.0	46.0	50.1
	SCOPE	54.2	58.0	56.1	45.9	49.1	47.4	72.0	71.7	71.8	61.3	61.1	61.2
	ECSpell	**75.8**	65.8	**70.4**	**67.6**	58.6	62.8	84.9	79.7	82.2	75.9	71.2	73.5
	ChatGPT	23.5	22.2	22.8	20.4	19.3	19.9	-	-	-	-	-	-
	RSpell$^{\mathbf{S}}$	52.0	62.0	56.5	45.5	54.3	49.5	71.6	74.8	73.2	65.3	68.1	66.7
	RSpell$^{\mathbf{E}}$	73.0	**67.5**	70.1	66.7	**61.7**	**64.1**	**89.6**	**80.1**	**84.6**	**86.1**	**77.0**	**81.3**
odw	BERT	79.8	62.6	70.1	74.0	58.1	65.1	85.3	74.9	79.8	78.8	69.2	73.7
	ReaLiSe	49.6	43.8	46.5	38.0	33.6	35.6	64.0	58.9	61.4	55.0	50.6	52.7
	SCOPE	82.6	73.4	77.7	75.7	67.3	71.3	88.4	81.0	84.5	82.2	75.3	78.6
	ECSpell	82.4	70.1	75.8	76.9	64.3	70.2	88.2	79.9	83.8	82.3	74.5	78.2
	ChatGPT	53.5	22.8	32.0	45.6	19.4	27.2	-	-	-	-	-	-
	RSpell$^{\mathbf{S}}$	84.6	**80.9**	**82.7**	77.3	**73.9**	**75.5**	90.1	**86.7**	**88.4**	83.4	**80.2**	81.8
	RSpell$^{\mathbf{E}}$	**87.4**	72.5	79.3	**80.6**	66.8	73.1	**92.4**	82.9	87.4	**89.0**	79.9	**84.2**

4.4 Main Results

Table 2 shows the sentence-level performance of RSpell and baseline methods on three domain datasets, law, medicine and official document writing, for both zero-shot and fine-tuning scenarios. With the help of retrieval framework, RSpell^{S} and RSpell^{E} outperform other baselines. Compared with the original spellers, our proposed RSpell framework has achieved at least 1.3% improvement under zero shot setting and 3–8% improvement under fine-tuning setting, respectively, which indicates that incorporating retrieval information is a highly effective approach for CSC tasks in domain-specific data. We can see that the proposed framework achieves a greater improvement under the fine-tuning setting compared to the zero-shot setting. We speculate that the most likely reason for this is that the model can better adapt to the new domain-specific data during fine-tuning, which also implies that there is still room for further improvement in the model's generalization ability. Additionally, we observe that despite the strong language capabilities of large language models, their output format is often unstable, making them less suitable for CSC tasks.

4.5 Ablation Studies

To explore the effectiveness of each component of RSpell, we conducted ablation studies with different settings: 1) removing the information retrieval module (w/o IR), 2) removing the adaptive process controller (w/o APC), and 3) removing the second iteration strategy (w/o SSS). As shown in Table 3, the performance drops regardless of which component is removed, proving the effectiveness of each component.

Table 3. Ablation results on test sets in three domains: law, medicine, and official document writing. The following modifications were made to RSpell^{E}: removing information retrieval (w/o IR), removing adaptive process controller (w/o APC), and remove the secondary search strategy (w/o SSS). It is worth noting that the absence of IR then includes the absence of APC and the absence of SSS. Best results are in **bold**.

Method	Law		Med		Odw	
	D-F	C-F	D-F	C-F	D-F	C-F
RSpell^{E}	**89.0**	**83.4**	**84.6**	**81.3**	87.4	**84.2**
w/o IR	84.0	76.7	79.7	74.1	82.0	74.7
w/o APC	87.7	82.1	83.4	78.7	**87.5**	83.5
w/o SSS	86.8	81.1	80.8	78.0	84.2	81.0

4.6 Effect of Varying Lexicons Sizes

Incorporating relevant external knowledge through retrieval can effectively assist the CSC model in error correction, and the degree of the external knowledge's impact is a key factor. We found that the size of the retrieval thesaurus has a significant impact on the experimental results. We use the word segmentation tool to extract key phrases from the corresponding training corpus and extend the three benchmark lexicons accordingly. As shown in Table 4, with the increase of the retrieval thesaurus size, the performance improvement becomes more apparent. This indicates that expanding the retrieval thesaurus can retrieve more effective information and thus improve the error correction effect.

Table 4. The performance impact of using retrieval thesaurus of different sizes in the fine-tuning scenario was compared on datasets in three domains: law, medicine, and official document writing. Best results are in **bold**.

Dataset	Lexicons Size	Detection Level			Correction Level		
		Pre	Rec	F1	Pre	Rec	F1
law	w/o Expanding(9896)	89.5	83.9	86.6	82.4	77.3	79.8
	w/ Expanding(33121)	**91.0**	**87.1**	**89.0**	**85.3**	**81.6**	**83.4**
med	w/o Expanding(18749)	89.6	80.1	84.6	84.7	75.7	79.9
	w/ Expanding(21583)	**89.6**	**80.1**	**84.6**	**86.1**	**77.0**	**81.3**
odw	w/o Expanding(12509)	92.3	82.5	87.2	88.1	78.7	83.1
	w/ Expanding(29778)	**92.4**	**82.9**	**87.4**	**89.0**	**79.9**	**84.2**

5 Conclusion

We propose RSpell, a retrieval-augmented framework for domain adaptive CSC. RSpell leverages retrieval methods to transform sentences into phonetic sequences based on the characteristics of the CSC task. It utilizes fuzzy matching to retrieve relevant external knowledge and guides the spell checker for accurate error correction. RSpell sets new benchmarks on three domain-specific CSC datasets, demonstrating that incorporating retrieval information is a highly effective approach for CSC tasks.

Acknowledgments. We thank all reviewers for their valuable comments. This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China (No. 62106165), the National Natural Science Foundation of China (No. 62076173).

References

1. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1870–1879 (2017)

2. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: Barzilay, R., Kan, M. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1870–1879 (2017)
3. Cheng, X., et al.: SpellGCN: incorporating phonological and visual similarities into language models for Chinese spelling check. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 871–881 (2020)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
5. Gu, J., Wang, Y., Cho, K., Li, V.O.K.: Search engine guided non-parametric neural machine translation. CoRR abs/1705.07267 (2017)
6. Gu, L., Wang, Y., Liang, X.: Introduction to NJUPT Chinese spelling check systems in CLP-2014 bakeoff. In: Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pp. 167–172 (2014)
7. Hong, Y., Yu, X., He, N., Liu, N., Liu, J.: FASPell: a fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm. In: Proceedings of the 5th Workshop on Noisy User-Generated Text, W-NUT@EMNLP 2019, pp. 160–169 (2019)
8. Huang, L., et al.: PHMOSpell: phonological and morphological knowledge guided Chinese spelling check. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 5958–5967 (2021)
9. Jiang, Y., et al.: A rule based Chinese spelling and grammar detection system utility. In: 2012 International Conference on System Science and Engineering (ICSSE), pp. 437–440. IEEE (2012)
10. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **60**(5), 493–502 (2004)
11. Kaneko, M., Takase, S., Niwa, A., Okazaki, N.: Interpretability for language learners using example-based grammatical error correction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 7176–7187 (2022)
12. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 6769–6781 (2020)
13. Kassner, N., Schütze, H.: BERT-kNN: adding a kNN search component to pretrained language models for better QA. In: Findings of the Association for Computational Linguistics, pp. 3424–3430 (2020)
14. Khandelwal, U., Fan, A., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Nearest neighbor machine translation. CoRR abs/2010.00710 (2020)
15. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: nearest neighbor language models. In: 8th International Conference on Learning Representations (2020)
16. Li, J., Wang, Q., Mao, Z., Guo, J., Yang, Y., Zhang, Y.: Improving Chinese spelling check by character pronunciation prediction: the effects of adaptivity and granularity. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 4275–4286 (2022)
17. Liu, C., Lai, M., Chuang, Y., Lee, C.: Visually and phonologically similar characters in incorrect simplified Chinese words. In: Huang, C., Jurafsky, D. (eds.) 23rd International Conference on Computational Linguistics, COLING 2010, Posters Volume, pp. 739–747. Chinese Information Processing Society of China (2010)

18. Liu, S., et al.: CRASpell: a contextual typo robust approach to improve Chinese spelling correction. In: Findings of the Association for Computational Linguistics, pp. 3008–3018 (2022)
19. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
20. Lv, Q., Cao, Z., Geng, L., Ai, C., Yan, X., Fu, G.: General and domain-adaptive Chinese spelling check with error-consistent pretraining. ACM Trans. Asian Low Resour. Lang. Inf. Process. **22**(5), 1–18 (2023)
21. Tseng, Y., Lee, L., Chang, L., Chen, H.: Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP, pp. 32–37 (2015)
22. Turnbull, D.: BM25 the next generation of Lucene relevance (2015). https://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-oflucene-relevation/. Accessed 21 Sept 2018
23. Wang, B., Che, W., Wu, D., Wang, S., Hu, G., Liu, T.: Dynamic connected networks for Chinese spelling check. In: Findings of the Association for Computational Linguistics, pp. 2437–2446 (2021)
24. Wang, D., Song, Y., Li, J., Han, J., Zhang, H.: A hybrid approach to automatic corpus generation for Chinese spelling check. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2517–2527 (2018)
25. Wang, Y., Liao, Y., Wu, Y., Chang, L.: Conditional random field-based parser and language model for traditional Chinese spelling checker. In: Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, pp. 69–73 (2013)
26. Weston, J., Dinan, E., Miller, A.H.: Retrieve and refine: improved sequence generation models for dialogue. In: Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP, pp. 87–92 (2018)
27. Wu, J., Chiu, H., Chang, J.S.: Integrating dictionary and web N-grams for Chinese spell checking. Int. J. Comput. Linguist. Chin. Lang. Process. **18**(4), 17–29 (2013)
28. Wu, S., Liu, C., Lee, L.: Chinese spelling check evaluation at SIGHAN bake-off 2013. In: Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP, pp. 35–42 (2013)
29. Xu, H., et al.: Read, listen, and see: leveraging multimodal information helps Chinese spell checking. In: Findings of the Association for Computational Linguistics, pp. 716–728 (2021)
30. Yu, J., Li, Z.: Chinese spelling error detection and correction based on language model, pronunciation, and shape. In: Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pp. 220–223 (2014)
31. Yu, L., Lee, L., Tseng, Y., Chen, H.: Overview of SIGHAN 2014 bake-off for Chinese spelling check. In: Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pp. 126–132 (2014)
32. Zhang, R., et al.: Correcting Chinese spelling errors with phonetic pre-training. In: Findings of the Association for Computational Linguistics, pp. 2250–2261 (2021)
33. Zhang, S., Huang, H., Liu, J., Li, H.: Spelling error correction with soft-masked BERT. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 882–890 (2020)
34. Zhang, S., Xiong, J., Hou, J., Zhang, Q., Cheng, X.: HANSpeller++: a unified framework for Chinese spelling correction. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, pp. 38–45 (2015)
35. Zhu, C., Ying, Z., Zhang, B., Mao, F.: MDCSpell: a multi-task detector-corrector framework for Chinese spelling correction. In: Findings of the Association for Computational Linguistics, pp. 1244–1253 (2022)

Exploiting Multiple Features for Hash Codes Learning with Semantic-Alignment-Promoting Variational Auto-encoder

Jiayang Chen[1] and Qinliang Su[1,2,3](✉)

[1] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
chenjy265@mail2.sysu.edu.cn

[2] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China

[3] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Beijing, China
suqliang@mail.sysu.edu.cn

Abstract. Semantic hashing is an effective technique to empower information retrieval. Currently, considerable efforts have been dedicated to generating high-quality hash codes by modeling document features using generative models and other approaches. However, most of these methods rely solely on a single type of feature, such as TFIDF features, BERT embeddings, etc. As different types of features have distinct but complementary information of documents, *e.g.* TFIDF mainly contains the keywords information and BERT focuses on the semantics, hash codes generated solely from either may not capture the full essence of the documents. To overcome this challenge, we propose a semantic-alignment-promoting variational auto-encoder to generate hash codes from multiple document features. Specifically, a VAE-based generative model is first developed to model the multiple features. Then, we propose a semantic-alignment-promoting inference network to estimate the parameters of the variational posterior from multiple features. Additionally, the quality of hash codes is further improved by promoting the semantic alignment between the hash codes of connected documents in a constructed connection graph. The results of extensive experiments on three public datasets demonstrate that our proposed model significantly outperforms current state-of-the-art models.

Keywords: Multiple Features · Generative Hashing · Semantic Alignment

1 Introduction

Similarity search aims to find the most similar items to a query from a large dataset, and one of its applications is document retrieval. However, the majority of document features are real-valued, which makes the similarity calculations between numerous document pairs computationally expensive and requires

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 563–575, 2023.
https://doi.org/10.1007/978-3-031-44693-1_44

extensive storage space. A common solution is semantic hashing, which compresses the real-valued features into compact but informative binary codes, substantially enhancing similarity calculation speed while decreasing storage space demands.

To generate high-quality hash codes under unsupervised settings, extensive efforts have been made. One popular approach is through unsupervised generative models, such as variational auto-encoder (VAE) [1]. By requiring the hash codes of documents to generate the original features, the rich semantics will be preserved in hash codes [2–5]. Besides generative models, recent studies have also learned effective hash codes based on mutual information maximization theory [6,7] or contrastive learning principle [8].

In previous semantic hashing methods, TFIDF has been widely adopted to represent a document [2–6]. Since TFIDF, a statistical measure of word frequency, is effective in identifying keywords, the hash codes derived from it can capture the semantics associated with keywords. However, TFIDF fails to capture the contextual information within the document, leading to the generated hash codes lack of expressing the context of the document. In recent years, the success of pre-trained models has prompted researchers [7,8] to explore learning hash codes based on BERT embeddings [9]. By leveraging the rich semantics within BERT embeddings, excellent retrieval accuracy was achieved. However, since BERT is designed to serve multiple downstream tasks, certain helpful information for document retrieval, such as document keywords, is not prominent in the BERT embedding. Therefore, as each type of feature has distinct but complementary information of the document, we are supposed to leverage their complementary advantages for higher-quality hash codes.

In this paper, we propose to learn high-quality hash codes from the multiple features of documents with a semantic-alignment-promoting variational auto-encoder. Specifically, we develop a VAE-based generative model to model the multiple types of features of documents simultaneously. To preserve as much semantics information from different types of features as possible in the hash codes, it is necessary to design an appropriate inference network, instead of simply using an MLP network to process the concatenation of multiple features, to estimate the variational posterior more accurately. To this end, independent encoders are first employed on the multiple features, extracting semantics information from various perspectives. These encoded features are subsequently combined via a fusion operator (*e.g.* a weighted summation) into the parameters of the variational posterior. However, due to the misalignment in semantics of the multiple encoded features, mixing them without calibration may lead to messy hash codes. To address the issue, we propose to promote the semantic alignment between the multiple encoded features by leveraging contrastive learning [10]. Specifically, the multiple encoded features of the same document are viewed as positive samples and the encoded features are thereby well regularized. In addition to the semantic alignment among the encoded features of the same document, a constructed connection graph is also incorporated to promote the alignment between the hash codes of similar documents. By regarding the hash codes of connected documents as positive samples as well, further improvement can be expected to be seen in the quality of hash codes. The results of extensive experiments on three public datasets demonstrate that our proposed model outperforms current state-of-the-art models.

2 Preliminaries

VAE-Based Generative Hashing. To maximally preserve the semantic information from TFIDF features of documents in hash codes, a normal way is modeling it with a VAE-based generative model and then the latent representations will be viewed as the learned hash codes. Specifically, given a corpus $\{1, 2, ..., N\}$, VAE-based generative hashing methods propose to model the TFIDF features x_i of document i with a latent-variable model, that is

$$p(x_i, z_i) = p_\theta(x_i|z_i)p(z_i), \tag{1}$$

where $p(z_i) = \text{Bernoulli}(z_i)$ is the prior distribution of latent variable $z_i \in \{0,1\}^d$; and $p_\theta(x_i|z_i)$ is the decoder. For TFIDF features, the decoder is often defined as a factorized form, which is

$$p_\theta(x_i|z_i) = \prod_{j=1}^{|x_i|} p_\theta(w_{ij}|z_i) = \prod_{j=1}^{|x_i|} \frac{\exp(z_i^T E w_{ij} + b_j)}{\sum_{k=1}^{|V|} \exp(z_i^T E w_{ik} + b_k)}, \tag{2}$$

and w_{ij} denoting a one-hot vector representing the j-th word of document i; $E \in \mathbb{R}^{d \times |V|}$ is a learnable embedding matrix connecting latent code z_i and one-hot representation of word w_{ij}; and b_j is the bias. After the evidence lower bound (ELBO) of the log-likelihood $\log p(x_i)$ is maximized, the learned latent representations of a document are the desired hash code.

Contrastive Learning. Contrastive learning is an effective method to learn representations by contrasting two views of data with some negative samples. Specifically, given a minibatch of data v_i for $i = 1, 2, ..., B$, two different views of data $v_i^{(1)}$ and $v_i^{(2)}$ are passed through an encoder $f_1(\cdot)$ to produce continuous representations, that is

$$r_i^{(k)} = f_1\left(v_i^{(k)}\right), \quad k = 1 \text{ or } 2. \tag{3}$$

Because $r_i^{(1)}$ and $r_i^{(2)}$ are two representations of the same data v_i, the same semantic information is expected to be shared between them. To achieve this effect, contrastive learning first project $r_i^{(1)}$ and $r_i^{(2)}$ into a new latent space, that is

$$h_i^{(k)} = f_2\left(r_i^{(k)}\right), \quad k = 1 \text{ or } 2, \tag{4}$$

and then minimize the contrastive loss (NT-Xent Loss) [10] on $h_i^{(k)}$ as $\mathcal{L}_{\mathrm{CL}} = \sum_{i=1}^{B}\left(l_i^{(1)} + l_i^{(2)}\right)$ with

$$l_i^{(1)} \triangleq -\log \frac{e^{\operatorname{sim}(h_i^{(1)}, h_i^{(2)})/\tau}}{\sum_{j=1}^{B} \mathbb{1}_{[j \neq i]}\left(e^{\operatorname{sim}(h_i^{(1)}, h_j^{(1)})/\tau} + e^{\operatorname{sim}(h_i^{(1)}, h_j^{(2)})/\tau}\right)}, \tag{5}$$

where $\mathbb{1}_{[j \neq i]}$ is an indicator function; τ is a temperature parameter; and $\operatorname{sim}(\cdot,\cdot)$ is the similarity function. When $\mathcal{L}_{\mathrm{CL}}$ is minimized, the encoder $f_1(\cdot)$ is capable of extracting informative representations from the original data for downstream applications.

3 Semantic-Alignment-Promoting VAE

As previously mentioned, there exist multiple approaches to represent a document, such as BOW, TFIDF, and BERT. Therefore, we propose a semantic-alignment-promoting variational auto-encoder capable of leveraging multiple features to yield hash codes enriched with richer semantic information. In this section, we first propose a basic VAE to model the multiple features, and subsequently further promote it with semantic alignment between multiple encoded features and the hash codes of connected documents.

3.1 VAE for Multiple Features

Assuming we can access K types of features of a corpus with N documents, denoted as $\mathcal{R} \triangleq \{\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^K\}$, we propose to jointly model these features with latent variables $Z \triangleq \{z_1, z_2, ..., z_N\}$ in a factorized form, that is

$$p(\mathcal{R}, Z) = \prod_{i=1}^{N} p(x_i^1, x_i^2, \ldots, x_i^K, z_i) = \prod_{i=1}^{N}\left(p(z_i)\prod_{k=1}^{K} p_{\theta_k}(x_i^k|z_i)\right), \tag{6}$$

where $\{p_{\theta_k}(x_i^k|z_i)\}_{k=1}^{K}$ are K decoders to generate different types of features from the latent variable z_i; and the prior distribution $\{p(z_i)\}_{i=1}^{N}$ of z_i is defined as Bernoulli distribution, which aligns our latent variable with the binary characteristics of hash codes, same as the definition of NASH in [3].

Since the characteristics of different types of features are diverse, such as binary and real-valued, the decoders should be distributions that conform to the characteristics. For instance, previous studies used a categorical distribution as the decoder for TFIDF features, that is $p_\theta(x_i|z_i) = \prod_{j=1}^{|x_i|} p_\theta(w_{ij}|z_i)$ with $p_\theta(w_{ij}|z_i)$ defined as (2) in preliminaries. For real-valued features, such as BERT embeddings, we set the decoder as an independent multivariate Gaussian distribution, that is $p_\theta(x_i|z_i) = \mathcal{N}(x_i; \mu_i, I_d)$, where μ_i is dependant on z_i.

To train the multiple features hashing model (6), the objective is to maximize the log-likelihood $\log p(\mathcal{R}) = \log \int p(\mathcal{R}, Z)dZ$. However, since the exact log-likelihood is intractable, we instead maximize its lower bound (ELBO) under the framework of variational inference, that is

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}} &= \mathbb{E}_{q_\phi(Z|\mathcal{R})}\left[\log p_\theta(\mathcal{R}|Z)\right] - KL\left(q_\phi(Z|\mathcal{R})\|p(Z)\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q_\phi(z_i|\mathcal{R}_i)}\left[\log p_{\theta_k}(x_i^k|z_i)\right] - \sum_{i=1}^{N} KL\left(q_\phi(z_i|\mathcal{R}_i)\|p(z_i)\right),\end{aligned} \tag{7}$$

where $q_\phi(Z|\mathcal{R})$ is the variational posterior. In this paper, we assume it to maintain a factorized Bernoulli distribution form, that is $q_\phi(Z|\mathcal{R}) = \prod_{i=1}^{N} q_\phi(z_i|\mathcal{R}_i)$ with $\mathcal{R}_i \triangleq \{x_i^1, x_i^2, \ldots, x_i^K\}$. Specifically, $q_\phi(z_i|\mathcal{R}_i)$ is defined as

$$q_\phi(z_i|\mathcal{R}_i) = \mathrm{Bernoulli}(z_i; p_i), \tag{8}$$

where $p_i \in (0,1)^d$ denotes the parameter of multivariate Bernoulli distribution. In our experiments, $p_i = \mathrm{Sigmoid}(g_\phi(\mathcal{R}_i))$ is the output of a inference network $g_\phi(\cdot)$, which is parameterized by ϕ. As the variational posterior $q_\phi(z_i|\mathcal{R}_i)$ conditioning on $\mathcal{R}_i$, it serves a role to fuse the diverse information provided by multiple features. To achieve information fusion, we design the inference network $g_\phi(\cdot)$ in a weighted summation form, that is

$$g_\phi(\mathcal{R}_i) = \sum_{k=1}^{K} w_k \cdot g_\phi^k\left(x_i^k\right), \tag{9}$$

where $g_\phi^k(\cdot)$ is the specific encoder to encode the k-th features and w_k is a hyperparameter to control its contribution to the final hash code.

To make the gradient pass through the latent variable z_i successfully, we first reformulate z_i with the reparametrization trick, that is $z_i = (\mathrm{sign}(p_i - \xi_i) + 1)/2$ with $\xi \sim \mathrm{U}(0,1)$ being a sample from the uniform distribution. Then, we adopt the straight-through gradient estimator in [11] to enable the gradients to pass through $\mathrm{sign}(\cdot)$. After training, the hash code of a document i can be obtained by binarizing p_i in a deterministic manner, *e.g.*, $z_i = 0.5 \times (\mathrm{sign}(p_i - 0.5) + 1)$.

3.2 Promoting Through Semantic Alignment

In the variational posterior of the multiple features hashing model, we employ a straightforward weighted summation to fuse the information from different types of features. Since the different features are encoded by K independent decoders $\{g_\phi^k(\cdot)\}_{k=1}^{K}$, the semantics of the multiple encoded features are not aligned. Simply adding them together would result in the semantics from different types of features being chaotically mixed up, leading to poor hash code quality. Hence, the semantically aligned encoded features from different types are desired to enhance the efficiency of fusion.

To achieve this goal, we propose to align different types of encoded features of one document with contrastive learning, by treating them as positive pairs. Denoting the encoded features as $\tilde{x}_i^k = g_\phi^k\left(x_i^k\right)$, we define the feature alignment contrastive loss between multiple encoded features as follows:

$$\mathcal{L}_{\mathrm{CL}}^{R} \triangleq -\sum_{i=1}^{N}\sum_{(x_i^{k_1}, x_i^{k_2}) \in \mathcal{R}_i} \log \frac{s\left(\tilde{x}_i^{k_1}, \tilde{x}_i^{k_2}\right)}{\frac{1}{2}\sum_{j\in\mathcal{B}}\left(s\left(\tilde{x}_i^{k_1}, \tilde{x}_j^{k_2}\right) + s\left(\tilde{x}_j^{k_1}, \tilde{x}_i^{k_2}\right)\right)}, \tag{10}$$

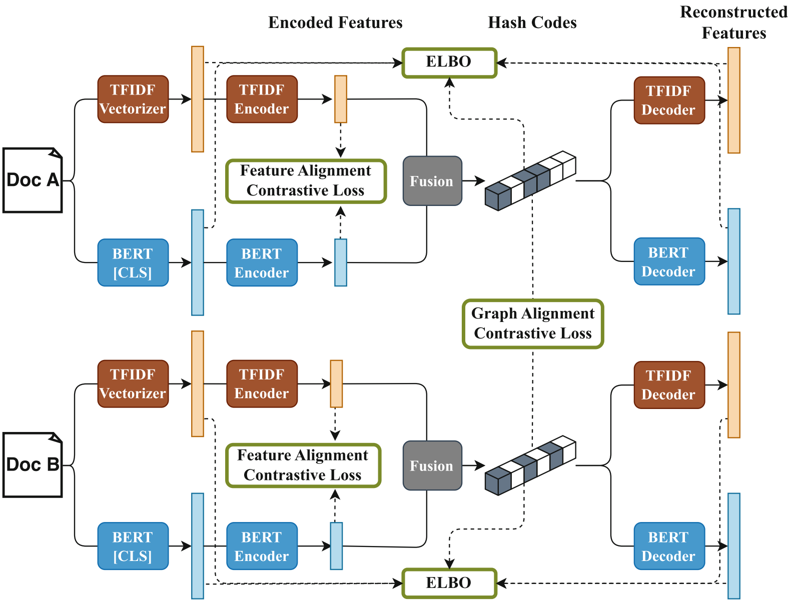


Fig. 1. A demonstration of the complete model architecture, assuming only TFIDF features and BERT embeddings are available for simplicity.

where $s(y_1, y_2) = \exp\left(y_1 \cdot y_2^\top / \tau\right)$ is the similarity function and $\mathcal{B}$ is the minibatch of documents when training. Additionally, we set $k_1 < k_2$ to avoid redundancy. As the contrastive loss $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ is minimized, the multiple encoded features of the same documents tend to cluster in adjacent regions of the same feature space, which implies that the semantically aligned encoded features are obtained.

In addition to requiring the multiple encoded features of one document to be semantically aligned, the semantic alignment between the hash codes of documents from the same category is also essential to improve the performance of hash codes. However, since the categories of documents are assumed to be unknown, there is no explicit constraint in the model to align the semantics of hash codes. Fortunately, we can partially achieve this goal by leveraging a connection graph constructed with the KNN algorithm on the raw features of documents. Specifically, we propose to perform contrastive learning on the hash codes of connected documents in the connection graph to align the semantics between them. By regarding the connected documents in the graph as positive samples pairs, we define the graph alignment contrastive loss as

$$\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}} \triangleq -\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \log \frac{s(z_i, z_j)}{s(z_i, z_j) + \sum_{n \in \mathcal{B} \setminus i} s(z_i, z_n)}, \tag{11}$$

where $\mathcal{N}_i$ denotes a set of documents that connecting to document i; $s(\cdot, \cdot)$ is the same similarity function defined in (10); and $\mathcal{B}$ is the minibatch of documents when training. By applying contrasting learning on the hash codes of connected documents, their similarity will be forced to be large, which implies that the goal of semantic alignment between the hash codes of documents from the same category is partially achieved, as connected documents are usually from the same category.

By integrating $\mathcal{L}_{\text{ELBO}}$, $\mathcal{L}_{\text{CL}}^{\text{R}}$ and $\mathcal{L}_{\text{CL}}^{\text{G}}$ through weighted summation, we can obtain the final objective function of our proposed model, that is

$$\mathcal{L} \triangleq \mathcal{L}_{\text{ELBO}} - \alpha \cdot \mathcal{L}_{\text{CL}}^{\text{R}} - \beta \cdot \mathcal{L}_{\text{CL}}^{\text{G}}. \tag{12}$$

When this objective function is maximized, we can obtain hash codes that integrate multiple features and the connection graph among documents. Figure 1 presents the complete architecture of our model, and for a simple architecture, we assume that only two views, TFIDF and BERT, are available. To facilitate subsequent discussion, we name our proposed model as **S**emantic-**A**lignment-Promoting **M**ultiple **F**eatures **H**ashing (SAMFH).

4 Related Works

Unsupervised semantic hashing has gained significant research attention in document retrieval. The common approach is based on generative models like variational autoencoder (VAE) [1] for learning high-quality hash codes. For instance, VDSH [2] models documents using VAE and generates hash codes through binarization. NASH [3] replaces the Gaussian prior in VAE with Bernoulli prior and leverages the straight-through technique [11] for end-to-end training. BMSH [12] and CorrSH [13] improve performance using more expressive distributions. WISH [14] tackles information loss in few-bit scenarios with auxiliary implicit topic vectors. In addition to learning hash codes solely from document contents, some of the recent works further improve the performance of hash codes by utilizing the connection graph among documents. NbrReg [4] and PairRec [5] required the hash code of a document to reconstruct the documents connected to it. RBSH [15] precisely captured the correlation between documents by using a ranking loss based on the graph.

In addition to generative hashing methods, recent studies have also explored learning hash codes based on mutual information maximization theory or contrastive learning principle. For example, AMMI [6] generated high-quality hash codes by maximizing the mutual information between document contents and hash codes. DHIM [7] learned the hash codes from the BERT embeddings by maximizing the mutual information between global codes and local codes from documents. CSH [8] sought to generate hash codes by contrasting the hash codes extracted from two BERT embeddings of the same document. Specifically, the two BERT embeddings were obtained by passing the same document into the pre-trained BERT model with dropout. However, these methods neglected the inclusion of multiple features of documents, leading to the inadequacy of the generated hash codes in fully representing the documents, since each type of feature has its own strengths and weaknesses. By addressing this problem with the semantic-alignment-promoting VAE for multiple features, we successfully further improve the performance of hash codes.

5 Experiments

5.1 Experiments Setup

Datasets. Three public datasets are used to evaluate the performance of our proposed model. The datasets are: 1) NYT [16], which consists of 11,527 news articles (train: 9,221, val: 1,154, test: 1,152) published by the New York Times and categorized into 26 categories; 2) AGNews [17], which contains 127,600 news articles (train: 114,839, val: 6,381, test: 6,380) gathered from academic news search engines and categorized into 4 categories; and 3) DBpedia [18], which includes 60,000 article abstracts (train: 50,000, val: 5,000, test: 5,000) from Wikipedia and categorized into 14 categories.

Baselines. Our proposed model will be compared with the following models: VDSH [2], NASH [3], BMSH [12], CorrSH [13], WISH [14], AMMI [6], PairRec [5], DHIM [7] and CSH [8]. The reported performances of all baselines are quoted from DHIM [7] and CSH [8] except PairRec, which is implemented in our environment.

Training Details. To fairly compare with the baselines, only TFIDF features and the [CLS] embeddings of pre-trained BERT are used. For hyperparameters, the learning rate is fixed to 0.005, the batch size is fixed to 64, and the temperature τ in $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$ is fixed to $\frac{d}{16}$ for all cases. According to the precision of the validation set, we select α from 1 to 100, β from 1 to 10 and the weights of different views $\{w_k\}_{k=1}^{K}$ from 1 to 10. Additionally, the connection graph is simply constructed with the 100-nearest documents based on the cosine similarity of TFIDF features and BERT embeddings. The Adam optimizer [19] with the default setting except learning rate is utilized to train the model.

Evaluation Metrics. To be consistent with previous works, retrieval precision is utilized as the evaluation metric. Specifically, we retrieve the top 100 documents that are most similar to a given query, based on the Hamming distance between hash codes. The retrieval precision is calculated as the proportion of retrieved documents that share the same label as the query. We report the average precision across all queries in the test set.

5.2 Performance Comparison with Baselines

We conducted extensive experiments on three public datasets to evaluate the performance of SAMFH. The results are presented in Table 1. Since the baselines only used TFIDF features or BERT embeddings, our model only uses these

Table 1. Overall performance comparison between proposed model and baselines. NASH$^+$ is NASH that takes the concatenation of TFIDF features and BERT embeddings as input. SAMFH$^-$ is SAMFH without $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$.

Method	NYT				AGNews				DBpedia			
	16b	32b	64b	128b	16b	32b	64b	128b	16b	32b	64b	128b
Using TFIDF features												
VDSH	68.77	68.77	75.01	78.49	67.32	67.42	72.70	73.86	67.79	72.64	78.84	84.91
NASH	74.87	75.52	75.08	73.01	65.74	69.34	72.72	74.33	78.02	79.84	79.79	76.76
WISH	70.15	70.03	64.48	68.94	74.53	74.79	75.05	72.70	82.82	82.76	82.10	78.22
BMSH	74.02	76.38	76.88	77.63	74.09	76.03	76.09	73.56	83.17	86.24	87.05	83.86
CorrSH	75.43	77.61	77.24	78.39	76.20	76.45	76.61	77.67	82.01	81.78	80.94	85.77
AMMI	71.06	76.48	77.37	78.03	76.47	76.61	77.32	78.23	84.51	89.53	90.78	91.03
PairRec	75.42	78.56	79.63	80.75	76.82	78.31	79.79	80.18	84.00	87.09	88.91	89.13
Using BERT [CLS] embeddings												
VDSH	53.38	58.18	62.44	64.64	62.97	66.35	69.57	70.27	69.59	75.21	79.54	80.62
NASH	55.87	58.25	60.98	64.27	66.32	68.44	70.40	72.07	65.87	74.54	77.96	81.43
WISH	58.83	64.75	65.47	70.34	65.35	66.19	69.39	72.03	65.65	72.91	76.66	82.29
BMSH	59.35	63.26	65.87	69.71	66.77	69.61	71.99	73.16	66.42	79.13	82.01	84.57
CorrSH	62.03	65.48	68.38	72.28	67.06	68.51	70.86	73.17	65.28	74.63	78.65	83.61
AMMI	60.47	65.10	69.67	74.47	65.50	68.26	71.85	74.36	80.25	82.67	89.26	86.74
PairRec	60.63	67.15	68.44	69.91	69.06	69.35	71.95	72.79	82.58	84.65	88.16	88.16
DHIM	79.69	80.55	79.77	79.09	78.23	79.17	78.88	79.86	94.26	94.80	93.02	88.21
CSH	79.63	81.05	81.08	81.28	79.14	80.25	81.14	81.78	95.19	95.84	95.79	96.01
Using TFIDF features and BERT [CLS] embeddings												
NASH$^+$	75.72	77.68	80.99	81.76	77.11	78.08	79.92	80.72	81.09	88.39	90.74	91.65
SAMFH$^-$	79.78	81.62	83.72	84.12	81.54	83.48	84.89	85.28	94.9	96.28	96.54	96.63
SAMFH	**84.11**	**85.61**	**86.99**	**87.39**	**82.63**	**84.14**	**85.00**	**85.62**	**96.14**	**97.09**	**97.39**	**97.49**

two views as inputs to ensure a fair comparison. The results clearly demonstrate that our model outperforms baselines that only use one type of feature with a substantial margin, highlighting the effectiveness of utilizing multiple features. Additionally, to verify the superiority of the designed semantically aligned inference network, we compare NASH$^+$, which is NASH that takes the concatenation of TFIDF features and BERT embeddings as input, with SAMFH$^-$, which is SAMFH without $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$. It is obvious that fusing the diverse information from multiple features by summing the semantically aligned encoded features is superior to simply inputting the concatenation of multiple features. Furthermore, the improvement from SAMFH$^-$ to SAMFH manifests the effectiveness of aligning the semantics of the hash codes of connected documents. Lastly, the performance improvement of SAMFH over PairRec partially suggests the superiority of utilizing the connection graph through contrastive learning on the hash codes of connected documents. Further experimental studies are also conducted to validate this point in the subsequent section.

5.3 Ablation Study on $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$

To investigate the impact of features alignment contrastive loss $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and graph alignment contrastive loss $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$, we conducted experiments with these losses removed, and the results are illustrated in Fig. 2. It is obvious that, across all three datasets, the hash codes generated by SAMFH exhibit the best retrieval accuracy for all bit lengths, indicating that $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$ can aid the model in producing higher quality hash codes. Additionally, upon scrutinizing the performance of SAMFH^{-}, SAMFH^{--} and NASH^{+}, we found that aligning the semantics of multiple encoded features before fusing them can effectively enhance the quality of hash codes. Without $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$, our proposed fusion method performs even worse than simply concatenating the multiple features. Finally, by comparing SAMFH and SAMFH^{-}, we discovered that the connection graph in NYT dataset can offer more performance gains than AGNews and DBpedia, implying that we must adjust the weight of $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$ in accordance with the attributes of different datasets to attain optimal performance.

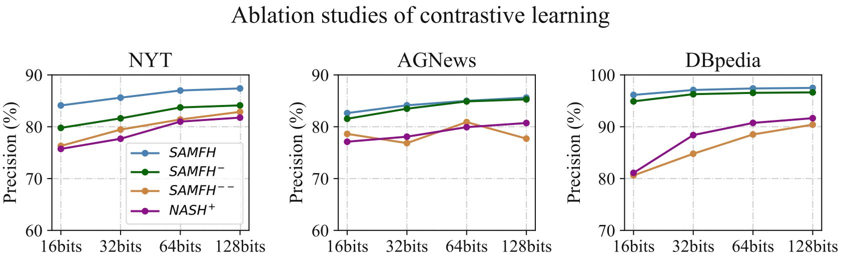


Fig. 2. Ablation studies to investigate the impact of $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$. The first presented model SAMFH^{--} is SAMFH without both $\mathcal{L}_{\mathrm{CL}}^{\mathrm{R}}$ and $\mathcal{L}_{\mathrm{CL}}^{\mathrm{G}}$.

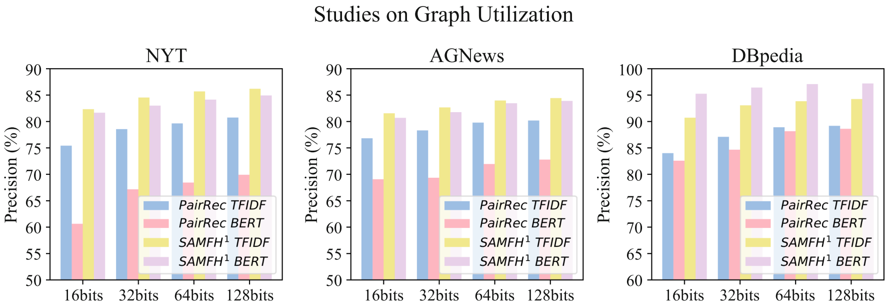


Fig. 3. Comparing PairRec with SAMFH^{1} (SAMFH only uses one type of features) to verify the effectiveness of leveraging connection graph by contrasting the hash codes of connected documents.

5.4 Studies on Connection Graph Utilization

To further validate the effectiveness of utilizing connection graph through contrastive learning on the hash codes of connected documents, SAMFH^1, which is our proposed model that only uses one type of view features, is compared with the baseline model PairRec and the experimental results are stated in Fig. 3. It is observed that SAMFH^1 with TFIDF or BERT consistently outperforms PairRec with TFIDF or BERT, indicating that leveraging connection graph by contrasting the hash codes of connected documents is a more effective method to preserve the graph information in hash codes space.

6 Conclusion

In this paper, we proposed semantic-alignment-promoting VAE that incorporates multiple features of documents into hash code generation. Specifically, We developed a VAE-based generative model to model multiple features simultaneously and aligned the semantics of multiple encoded features of the same document. Additionally, we further improve the quality of hash codes by leveraging the connection graph with contrastive learning. Excellent experimental results on three public datasets demonstrated the effectiveness of our approach.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62276280, U1811264), Key R&D Program of Guangdong Province (No. 2018B010107005), Natural Science Foundation of Guangdong Province (No. 2021A1515012299).

References

1. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings (2014)
2. Chaidaroon, S., Fang, Y.: Variational deep semantic hashing for text documents. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 75–84. SIGIR 2017. Association for Computing Machinery, New York, NY, USA (2017)
3. Shen, D., et al.: NASH: toward end-to-end neural architecture for generative semantic hashing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2041–2050. Association for Computational Linguistics, Melbourne, Australia, July 2018
4. Chaidaroon, S., Ebesu, T., Fang, Y.: Deep semantic text hashing with weak supervision. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, pp. 1109–1112. Association for Computing Machinery, New York, NY, USA (2018)
5. Hansen, C., Hansen, C., Simonsen, J.G., Alstrup, S., Lioma, C.: Unsupervised semantic hashing with pairwise reconstruction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, pp. 2009–2012. Association for Computing Machinery, New York, NY, USA (2020)

6. Stratos, K., Wiseman, S.: Learning discrete structured representations by adversarially maximizing mutual information. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 9144–9154. PMLR, 13–18 July 2020
7. Ou, Z., Su, Q., Yu, J., Zhao, R., Zheng, Y., Liu, B.: Refining BERT embeddings for document hashing via mutual information maximization. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, Punta Cana, Dominican Republic, November 2021
8. Qiu, Z., Su, Q., Yu, J., Si, S.: Efficient document retrieval by end-to-end refining and quantizing BERT embedding with contrastive product quantization. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 853–863. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, December 2022
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR, 13–18 July 2020
11. Bengio, Y., Léonard, N., Courville, A.C.: Estimating or propagating gradients through stochastic neurons for conditional computation. CoRR abs/1308.3432 (2013)
12. Dong, W., Su, Q., Shen, D., Chen, C.: Document hashing with mixture-prior generative models. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5226–5235. Association for Computational Linguistics, Hong Kong, China, November 2019
13. Zheng, L., Su, Q., Shen, D., Chen, C.: Generative semantic hashing enhanced via Boltzmann machines. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 777–788. Association for Computational Linguistics, Online, July 2020
14. Ye, F., Manotumruksa, J., Yilmaz, E.: Unsupervised few-bits semantic hashing with implicit topics modeling. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2566–2575. Association for Computational Linguistics, Online, November 2020
15. Hansen, C., Hansen, C., Simonsen, J.G., Alstrup, S., Lioma, C.: Unsupervised neural generative semantic hashing. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 735–744. SIGIR 2019. Association for Computing Machinery, New York, NY, USA (2019)
16. Tao, F., et al.: Doc2Cube: allocating documents to text cube without labeled data. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 1260–1265 (2018). https://doi.org/10.1109/ICDM.2018.00169
17. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc. (2015)

18. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web J. **6**(2), 167–195 (2015)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)

Enhancing Detailed Feedback to Chinese Writing Learners Using a Soft-Label Driven Approach and Tag-Aware Ranking Model

Yuzhe Cai[1], Shaoguang Mao[2(✉)], Chenshuo Wang[3], Tao Ge[2], Wenshan Wu[2], Yan Xia[2], Chanjin Zheng[4], and Qiang Guan[1]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
{caiyuzhe2022,qiang.guan}@ia.ac.cn
[2] Microsoft Reaserch Asia, Beijing, China
{shaoguang.mao,tage,wenswu,yanxia}@microsoft.com
[3] Peking University, Beijing, China
iven@ivenwang.com
[4] Department of Educational Psychology, East China Normal University, Shanghai, China
chjzheng@dep.ecnu.edu.cn

Abstract. This paper focuses on providing detailed and specific feedback for Chinese writing learners, which is challenging due to the uncertainty of the feedback space. 30 common tags are identified based on clustering 363k comment phrases. By predicting the corresponding tags for an input, learners can gain insights on how to improve their work. However, the various tag types and non-exhaustive annotation pose challenges for model training. To address this, we propose a soft-label-driven approach to construct a more accurate relationship between samples and the tag space. We use a relevance matrix between different tags to adjust the positive label's confidence across the entire tag space. Additionally, we propose a tag-aware regression-based ranking model that further improves performance. Our experiments demonstrate that the proposed soft-label approach performs better than a hard-label approach, and the proposed ranking model enhances performance. Our data and code are available at: https://github.com/Zhe0311/DetailedFeedback2ChineseWritingLearners.

Keywords: Automated feedback · Computer-aided language learning · Soft-label tag-aware ranking model · Essay assessment

1 Introduction

Providing detailed feedback to writing learners is crucial for their learning. Extensive researches have been conducted on automated evaluation systems, with much of the focus being on scoring, e.g., automated essay scoring

Y. Cai—Work was done when interning at Microsoft Research Asia.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 576–587, 2023.
https://doi.org/10.1007/978-3-031-44693-1_45

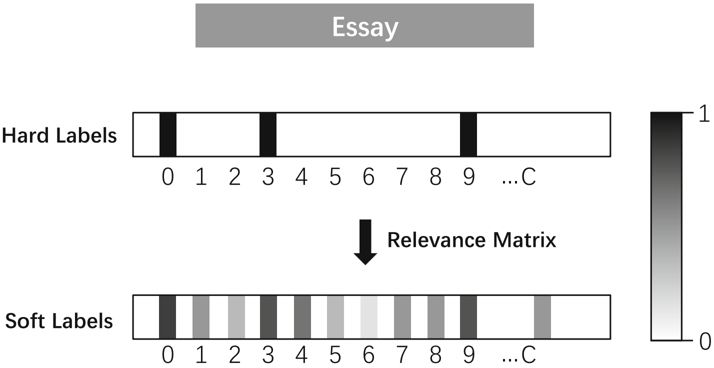


Fig. 1. The transformation from hard labels to soft labels. Each essay is manually annotated several tags as detailed feedback, but, unfortunately, the annotation is non-exhaustive. The incompletely annotated hard labels are transferred to soft labels through processing with a relevance matrix.

(AES) [8,11,12,14,21,23]. While providing detailed feedback, which can offer more guidance to language learners, remains challenge.

Automated feedback is typically considered a generation task that aims to generate comments for the writing learner's input. Recent advances in deep learning, specifically end-to-end neural networks based on pre-trained language models, have shown impressive results [3,7,19,20,22]. For a language model, writing fluent and readable text is easy, while identifying the weaknesses and strengths in students' writings is difficult. Even though the feedback by generative models is fluent and readable, it still cannot be specific and valuable for learners. In this paper, we focus on identifying the characteristics of students' writings.

We obtain common tags by clustering on a large number of teacher comments on Chinese native language learners' essays. The tags point out the strengths and weaknesses of essays from various perspectives such as logic, structure, content, expression, so on and so forth. A detailed and specific feedback is provided by assigning appropriate tags to essays.

Intuitively, the task can be viewed as a simple multi-task classification problem: given an input, make prediction on each tag, and return the corresponding tags. However, there are some challenges:

1. **Large output space and label sparsity issue.** There are totally more than one million raw comment phrases that we collected from teachers, so the output space is extremely large if all phrases are used as tags directly. Furthermore, the data exhibits a long-tail distribution, with limited training instances for most of the tags, leading to severe label sparsity issue.
2. **Non-exhaustive annotation.** Due to the large output space, teachers usually point out some of characteristics of an essay, whereas not including all the applicable tags. So, the manually annotated tags are not exhaustive.
3. **Lack of applicable metrics.** Since the data is under non-exhaustive annotation, conventional metrics for classification are not suitable for this task.

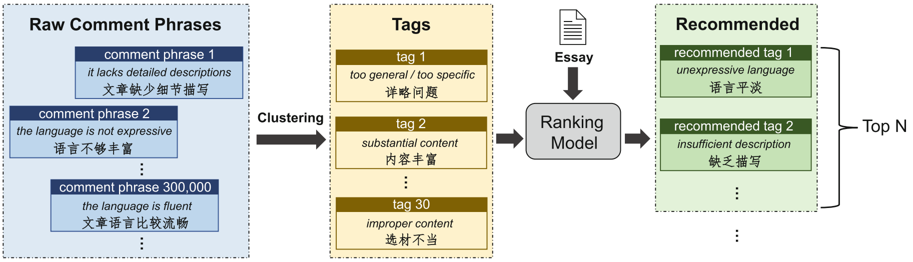


Fig. 2. The proposed ranking method framework. The large amount of raw comment phrases are clustered into 30 tags according to their semantic distance. Then the tags are sorted by the ranking model we propose. Finally, the top N tags are recommended.

There is a task known as Positive-Unlabeled (PU) learning in the field of machine learning, which is aimed to address the challenge of working with solely positive examples and unlabeled data [1,4,9,15]. There are some similarities between the non-exhaustive annotation challenge and the PU learning, as only part of the tags are pointed out to be positive while the rest remain unlabeled. However, the tags utilized in this research is inter-related. Through integrating the inspiration derived from PU learning with the specific characteristics of this task, the associations between tags are leveraged to better characterize the feedback tag space.

As shown in Fig. 2, in this paper, we first cluster 30 comment patterns from a large number of teacher comments to reduce the output space. Then the hard annotation is transferred to soft annotation through the proposed relevance matrix. We also propose a regression-based ranking model based on soft annotation. Finally, we propose a new metric mean maximum relevance (MMR) to better evaluate the model under the situation of non-exhaustive annotation setting. Experiments show that the proposed soft-label method can better represent the label distribution. And the proposed tag-aware regression-based model outperforms the multi-task method and KL-divergence-based ranking model.

2 Methods

2.1 Clustering Based on Semantic Distance

The extremely large size of comment phrases set presents a challenge of label sparsity. Nonetheless, many comment phrases indicate the equivalent semantic meaning with diverse expressions. So, the application of clustering based on semantic similarity reduces the output space, and mitigates the overlapping issue among the comment phrases.

All comment phrases are mapped to a 768-dimensional dense vector space with a sentence-transformers model[1] and then clustered into C clusters using

[1] https://huggingface.co/symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli.

K-means method. The process of clustering is followed by a meticulous manual merging and selection procedure conducted by a senior language teacher. After processing, there are 30 tags, and the teacher assigns a representative name to each tag. The 30 tags are listed in Table 1.

2.2 Relevance Matrix and Soft Labels

There are two characteristics of these essay-tags annotations: (1) the tags are inter-related. For example, if an essay is tagged as "insufficient description", it is likely to exhibit qualities of "unexpressive language" as well, and it is highly unlikely to possess the attribute of "vivid writing"; (2) the annotation is non-exhaustive, which also leads to a data imbalance issue.

To better utilize the non-exhaustively annotated data, it is transferred to soft labels, as shown in Fig. 1, which are believed to be a more accurate approximation of the theoretical true labels.

Relevance Matrix. A relevance matrix $\mathbf{M} \in \mathbb{R}^C$ is well-designed by language teaching experts. The value of $\mathbf{M}_{ij} \in [-10, 10]$ serves as a measure of the correlation between the i-th tag and j-th tag, where positive values represent relevance, and negative values indicate contradictions. In instances where two tags are completely unrelated, $\mathbf{M}_{ij}$ is set to 0.

Soft Labels. For a given essay $\mathbf{x}$, denote its hard annotation as $\mathbf{y}_h = [\mathbf{y}_h^{(1)}, \mathbf{y}_h^{(2)}, \ldots, \mathbf{y}_h^{(C)}]$, where $\mathbf{y}_h^{(i)} \in \{0, 1\}$ stands for the i-th tag, and denote its soft annotation as $\mathbf{y}_s = [\mathbf{y}_s^{(1)}, \mathbf{y}_s^{(2)}, \ldots, \mathbf{y}_s^{(C)}]$. $\mathbf{y}_s$ is derived from $\mathbf{y}_h$, and also leverages the correlation information in the relevance matrix. Essentially, a positive-annotated tag in $\mathbf{y}_h$ will result in the tag in $\mathbf{y}_s$ increased if the two exhibit a positive correlation, or decreased if a negative correlation is exhibited. The $\mathbf{y}_s$ is given by

$$\hat{\mathbf{y}}_s^{(i)} = \mathbf{y}_h^{(i)} + \sum_{k \neq i} \mathbb{I}\left(\mathbf{y}_h^{(k)} = 1\right) \alpha \mathbf{M}_{ik} \tag{1}$$

$$\mathbf{y}_s = \mathrm{softmax}(\hat{\mathbf{y}}_{\mathrm{s}}) \tag{2}$$

where $\mathbb{I}(\cdot)$ is an indicator function and $\alpha \in \mathbb{R}$ is a constant. Then, $\hat{\mathbf{y}}_s$ is normalized by a softmax function to construct a distribution. It should be noticed that the soft annotation ensures every tag is endowed with a supervised signal for all essays, thereby mitigating the label sparsity issue.

2.3 Ranking Model Based on KL-divergence

The Kullback-Leibler (KL) divergence measures the distance between two distributions and is widely utilized in the field of machine learning [6,10,17]. Given that soft labels are essentially distributions, the KL-divergence is well-suited to

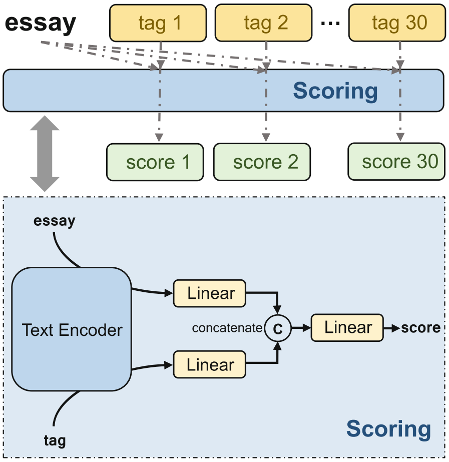


Fig. 3. The Tag-aware ranking model framework. Every essay-tag pair is regressed to a score by a fine-tuned BERT model. Then the tags can be sorted by their scores.

measure the distance between the soft labels and the model predictions in this study. For the given dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_{s,i})\}_{i=1}^{N}$, where $\mathbf{x}_i$ stands for the i-th essay and $\mathbf{y}_{s,i}$ denotes its corresponding soft labels, we hope to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}_s$ using deep neural networks, i.e., $f(\mathbf{x}_i) = \hat{\mathbf{y}}_i$, which maps essay space to tag space.

We adopt the BERT model as the input encoder, and adjust part of its parameters to optimize the performance on predicting distribution over the tags. Specifically, the 768-dimensional special token [CLS] is mapped to C dimensions through a fully connected layer, and then normalized by the softmax function. The minimization objective during training is given by

$$\min \sum_{i=1}^{N} D_{\mathrm{KL}} \left(\mathbf{y}_{s,i} \| \hat{\mathbf{y}}_i \right) \tag{3}$$

where the loss function $D_{\mathrm{KL}}(\cdot)$ is KL-divergence. In the inference stage, output for all tags are sorted in a descending order according to $\hat{\mathbf{y}}_i$, with the top n ($n < C$) tags being assigned to the input essay.

2.4 Tag-Aware Ranking Model

It is challenging for the KL-divergence ranking model (Sect. 2.3) to learn the mapping function due to the limited input information. We propose to utilize the tag information to enhance the task. The problem is viewed as learning a score function $f_{score} : \mathcal{X} \times \mathcal{Y}_s \rightarrow \mathbb{R}$, which should be optimized such that the appropriate tags have higher scores, whereas inappropriate tags have lower scores.

As shown in Fig. 3, the essay and labels are encoded jointly by a BERT model. The integrated information is leveraged to predict a score for the essay on the corresponding tag. One instance $(\mathbf{x}_i, \mathbf{y}_{s,i})$ is split into C instances: $(\mathbf{x}_i, \mathbf{t}^{(1)}, \mathbf{y}_{s,i}^{(1)})$, $(\mathbf{x}_i, \mathbf{t}^{(2)}, \mathbf{y}_{s,i}^{(2)})$, ..., $(\mathbf{x}_i, \mathbf{t}^{(C)}, \mathbf{y}_{s,i}^{(C)})$, where $\mathbf{t}^{(k)}$ denotes all the comment phrases corresponding to the k-th tag. The utilization of tag information directs the model to place a heightened focus on the perspective as depicted by the tag, leading to an improvement in performance. The loss function is

$$\mathcal{L} = \frac{1}{NC} \sum_{i=1}^{N} \sum_{j=1}^{C} \left(\mathbf{y}_{s,i}^{(j)} - f_{score} \left(\mathbf{x}_i, \mathbf{t}^{(j)} \right) \right)^2 \tag{4}$$

In the inference stage, the tags are sorted in a descending order according to output scores, with the top n $(n < C)$ tags being assigned to the input essay.

3 Experiments

3.1 Dataset

We collected 44, 904 Chinese essays from an online learning platform LeLeKetang[2]. All essays are along with teacher comments, covering various topics and skill levels. All comments are sliced and then filtered based on frequency, as the low-frequency comment phrases may be more related to the content details of essays. There are more than 363k comment phrases after filtering, which are all general comments on the strengths, weaknesses, and characteristics of the essays. Finally, the comment phrases are clustered into 30 clusters based on semantic distance (Sect. 2.1), as shown in Table 1. The 30 tags represent common patterns of feedback for Chinese writing. When applied to other languages, modifications of the tags can be made based on the characteristics of the specific language.

The data are publicly available for the re-implementation and further research.

- **44,904 instances** (35,000 for training, 5,000 for validation, and 4,904 for testing), each containing a Chinese essay and its annotation $\mathbf{y}_h \in \{0, 1\}^{30}$ on 30 tags.
- **Tag list**, listing the 30 tags and including all the comments phrases for each tag and its representative name.
- **Relevance matrix**, listing the correlation between tags (Sect. 2.2).

[2] http://www.leleketang.com/zuowen/.

Table 1. The 30 tags in both English and Chinese.

ID	Tags (EN)	Tags (CN)
1	unexpressive language	语言平淡
2	too general/ too specific	详略问题
3	unclear main idea	中心不突出
4	insufficient description	缺乏描写
5	inaccurate expressions	表达不准确
6	vivid writing	文笔生动
7	organized writing	叙事条理
8	substantial content	内容丰富
9	good writing flow	节奏鲜明
10	authentic language	语言自然
11	fluent language	语言流畅
12	well-structured writing	结构完整
13	improper content	选材不当
14	lacking in rhetorics	缺乏文采
15	clear arrangement of ideas	层次分明
16	unclear structure	结构不清晰
17	sincere emotion	情感真挚
18	unexpressive description	描写不生动
19	vivid language	语言生动
20	deficient in emotion	缺乏情感
21	literary interest	文学趣味
22	elegant language	语言优美
23	non-fluent language	语言不流畅
24	creative content	想象丰富
25	succinct language	语言简洁
26	improper main idea	立意不当
27	vivid description	描写生动
28	lacking in details	叙述不具体
29	stimulating reading interests	激发阅读兴趣
30	opening with the main idea	开篇点题

3.2 Metrics

Due to the non-exhaustive annotation, conventional metrics for classification fail to fully reflect the performance of the model. If a tag is annotated as negative, but is predicted to be positive, it is probably as the tag may actually be ignored during annotation. The lack of applicable metrics is also a challenge.

Mean Maximum Relevance. To address this problem, we propose a new metric Mean Maximum Relevance (MMR), where the interrelation across tags is taken to better determine the gain or loss of a prediction. Specifically, the gain or loss is determined by the correlation between this tag with those positive-annotated tags. For an essay $\mathbf{x}_i$, whose k-th tag is predicted to be positive, the maximum relevance (MR) of this tag is defined as

$$\mathrm{MR}\left(k, \mathbf{y}_{h,i}\right) = \max_{1 \leqslant j \leqslant 30,\ \mathbf{y}_{h,i}^{(j)}=1} \mathbf{M}_{kj} \tag{5}$$

For example, if an essay is annotated as “unexpressive language (1-st tag in 30 tags)” and “insufficient description (4-th tag in 30 tags)”. The MR of a prediction “inaccurate expressions (5-th tag in 30 tags)” is given by

$$\mathrm{MR} = \max\left\{\mathbf{M}_{1,5}, \mathbf{M}_{4,5}\right\} = \max\left\{5, 8\right\} = 8 \tag{6}$$

and the MR of prediction “vivid writing (6-th tag in 30 tags)” is

$$\mathrm{MR} = \max\left\{\mathbf{M}_{1,6}, \mathbf{M}_{4,6}\right\} = \max\left\{-10, -9\right\} = -9 \tag{7}$$

For all 30 tags, denote the prediction of $\mathbf{x}_i$ as $\hat{\mathbf{y}}_i$, the MMR of $\mathbf{x}_i$ is defined as

$$\mathrm{MMR}\left(\hat{\mathbf{y}}_i, \mathbf{y}_{h,i}\right) = \frac{1}{C} \sum_{k=1}^{C} \mathbb{I}\left(\hat{\mathbf{y}}_i^{(k)} = 1\right) \mathrm{MR}\left(k, \mathbf{y}_{h,i}\right) \tag{8}$$

The MMR of multiple instances is calculated as the average of their individual MMR. A larger MMR indicates a more consistent prediction with the raw information, and better model performance. MMR takes the correlation between the annotated categories and other categories into consideration, fully reflecting the relevance between the predicted and annotated tags, making it an effective evaluation metric for similar non-exhaustive annotation problem.

Other Metrics. Precision, recall and F0.5 are also reported for reference. F0.5 is adopted due to the greater priority of precision over recall. Precision@k and Recall@k are widely utilized in the extreme multi-label classification (XMC) problem [2,5,13,18], a task that addresses the challenges caused by an extremely large label set. We focus on top predictions by presenting MMR@k, Precision@k, Recall@k, and F0.5@k.

3.3 Experimental Setup

There are four groups of experiments: random guessing, multi-task approach, ranking model based on KL-divergence, and tag-aware ranking model.

1. **Random Guessing.** Given that task difficulty can vary depending on the number of tags, and that the MMR metric is biased for different sets of tags, a random guessing approach is utilized to indicate models' lower bound, while also reflecting the numerical characteristics of the metrics in settings of 10, 20, and 30 tags.
2. **Multi-task Approach.** A multi-task approach is adopted as a *baseline method*, i.e., performing binary classification on each tag for input. A BERT is fine-tuned with our data to minimize the cross-entropy loss between the predictions and tags. All the classifiers are trained jointly. Particularly, due to the data imbalance issue, the loss for every tag is weighted according to the proportion of positive to negative instances.
3. **Ranking Model Based on KL-divergence.** We conduct experiments on method in Sect. 2.3 to verify the advantages of soft labels. Rather than using hard labels, the KL-divergence loss is used to approximate the soft labels.
4. **Tag-aware Ranking Model.** We conduct experiments on method in Sect. 2.4 to validate the effectiveness of tag-aware approach.

Size of Tag Space. The 30 tags are sorted in a descending order according to frequency. To observe the robustness of the proposed method, we contrast the model performance in different size of the tag space. The number of tags are set as 10, 20, and 30, where only corresponding tags will be considered.

Number of Predicted Tags (@k). The number of predicted tags are determined manually. We focus on top k predictions where the k is set to 3 and 5, as it is reasonable to provide language learners 3–5 feedback tags in practical applications.

3.4 Experimental Results

The results shown in Table 2 demonstrate that:

1. The multi-task approach exhibits a high recall and low precision, primarily due to the absence of manual setting for the number of predicted tags. Every positive prediction made by model counts. So, it retrieves much more tags than other methods. Additionally, the low MMR further indicates the low relevance between the predicted tags and the annotated tags. The results collectively indicate that utilizing hard labels directly is an inadequate approach for this task.
2. The comparison of KL-divergence-based and multi-task methods indicates that significant improvement can be achieved through the utilization of soft labels. Furthermore, the introductions of tag-aware methods further contribute to the enhancements.

Table 2. Performance of different methods. MMR is ranging from −10 to 10. We present the result of making predictions randomly, denoted as Random, and a baseline system, denoted as Multi-task. Specifically, the performance of multi-task method on @3 and @5 are the same because the number of positive predictions is not determined manually. $\mathrm{RM}_{\mathrm{KL}}$ denotes KL-divergence-based ranking model (Sect. 2.3) and $\mathrm{RM}_{\mathrm{tag}}$ denotes tag-aware ranking model (Sect. 2.4).

Method	# of tags	MMR@3	Precision@3(%)	Recall@3(%)	F0.5@3(%)
Random	10	2.76	20.51	30.01	21.90
	20	3.91	13.69	15.00	13.93
	30	3.83	10.35	10.00	10.28
Multi-task	10	4.17	30.99	**66.88**	34.71
	20	4.62	24.13	**55.27**	27.19
(Baseline)	30	4.52	20.49	**66.46**	23.78
$\mathrm{RM}_{\mathrm{KL}}$	10	4.32	26.36	39.32	28.22
	20	5.98	31.30	34.13	31.82
	30	5.76	29.62	28.55	29.40
$\mathrm{RM}_{\mathrm{tag}}$	10	**4.96**	**34.41**	46.80	**36.33**
	20	**6.12**	**34.55**	37.09	**35.02**
	30	**5.89**	**31.32**	30.57	**31.17**

(a) Number of predicted tags k set to 3

Method	# of tags	MMR@5	Precision@5(%)	Recall@5(%)	F0.5@5(%)
Random	10	2.77	20.50	50.00	23.24
	20	3.91	13.69	24.99	15.05
	30	3.83	10.36	16.68	11.21
Multi-task	10	4.17	**30.99**	66.88	**34.72**
	20	4.62	24.13	**55.27**	27.19
(Baseline)	30	4.52	20.49	**66.46**	23.78
$\mathrm{RM}_{\mathrm{KL}}$	10	3.96	25.12	62.45	28.53
	20	5.73	25.08	45.57	27.56
	30	5.56	24.91	40.02	26.94
$\mathrm{RM}_{\mathrm{tag}}$	10	**4.61**	30.76	**69.73**	34.63
	20	**5.78**	**28.18**	50.43	**30.91**
	30	**5.76**	**26.29**	42.77	**28.49**

(b) Number of predicted tags k set to 5

3. As the size of tag space varies, the tag-aware ranking model keeps superior performance in comparison to other methods, verifying its high robustness.

4. As the number of recommended tags increases (from 3 to 5), the recall of all methods increases significantly. However, their MMR, precision, and F0.5-score decrease. Therefore, an appropriate k is crucial for practical application. The MMR and the other metrics can effectively serve as a reference for determining a suitable recommended tags number.

4 Discussion

The ChatGPT by OpenAI has caught widespread attention for its powerful capability [16]. However, some pilot experiments show it cannot achieve satisfying results on finding the advantages and disadvantages of the article. One promising way to better utilize ChatGPT is to combine the model we proposed with it as a two-staged pipeline. Several tags representing the strengths and weaknesses of a response are predicted first, and then are integrated into prompts to generate a more readable comment. The predicted tags guide the ChatGPT to generate comments in a right and specific direction. The comments generated in this way are more valuable and friendly to language learners.

5 Conclusions

We propose a soft-label approach and a tag-aware regression-based ranking model to provide detailed feedback to Chinese writing learners by utilizing tags obtained from over 363k comment phrases. Moreover, we introduce a novel metric that considers the correlation among labels to better conduct evaluations in this setting. This metric can also be applied to similar multi-label tasks where the labels are non-exhaustive but interrelated. Our experimental results demonstrate the superiority of the soft-label approach over a hard-label approach and the effectiveness of the proposed ranking model in enhancing performance.

References

1. Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey. Mach. Learn. **109**, 719–760 (2020)
2. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
3. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
4. Cao, N., Zhang, T., Shi, X., Jin, H.: Posistive-unlabeled learning via optimal transport and margin distribution
5. Chang, W.C., Yu, H.F., Zhong, K., Yang, Y., Dhillon, I.S.: Taming pretrained transformers for extreme multi-label text classification. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3163–3171 (2020)

6. Das Gupta, M., Srinivasa, S., Antony, M., et al.: Kl divergence based agglomerative clustering for automated vitiligo grading. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2700–2709 (2015)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Dong, F., Zhang, Y., Yang, J.: Attention-based recurrent convolutional neural network for automatic essay scoring. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 153–162 (2017)
9. Garg, S., Wu, Y., Smola, A.J., Balakrishnan, S., Lipton, Z.: Mixture proportion estimation and PU learning: a modern approach. Adv. Neural. Inf. Process. Syst. **34**, 8532–8544 (2021)
10. Goldberger, J., Gordon, S., Greenspan, H., et al.: An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In: ICCV, vol. 3, pp. 487–493 (2003)
11. Gong, J., et al.: IFlyEA: a Chinese essay assessment system with automated rating, review generation, and recommendation. In: Joint Conference on Natural Language Processing: System Demonstrations, pp. 240–248. Association for Computational Linguistics, Online, August 2021. https://doi.org/10.18653/v1/2021.acl-demo.29, https://aclanthology.org/2021.acl-demo.29
12. Hussein, M.A., Hassan, H., Nassef, M.: Automated language essay scoring systems: a literature review. PeerJ Comput. Sci. **5**, e208 (2019)
13. Jain, H., Prabhu, Y., Varma, M.: Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 935–944 (2016)
14. Ke, Z., Ng, V.: Automated essay scoring: a survey of the state of the art. In: IJCAI, vol. 19, pp. 6300–6308 (2019)
15. Kiryo, R., Niu, G., Du Plessis, M.C., Sugiyama, M.: Positive-unlabeled learning with non-negative risk estimator. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
16. OpenAI: ChatGPT (2022). https://openai.com/blog/chatgpt
17. Ponnoprat, D.: Differential privacy of Dirichlet posterior sampling. CoRR abs/2110.01984 (2021). https://arxiv.org/abs/2110.01984
18. Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., Varma, M.: Parabel: partitioned label trees for extreme classification with application to dynamic search advertising. In: Proceedings of the 2018 World Wide Web Conference, pp. 993–1002 (2018)
19. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
21. Ramesh, D., Sanampudi, S.K.: An automated essay scoring systems: a systematic literature review. Artif. Intell. Rev. **55**(3), 2495–2527 (2022)
22. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
23. Wang, Y., Wei, Z., Zhou, Y., Huang, X.J.: Automatic essay scoring incorporating rating schema via reinforcement learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 791–797 (2018)

Oral: Question Answering

NAPG: Non-Autoregressive Program Generation for Hybrid Tabular-Textual Question Answering

Tengxun Zhang[1], Hongfei Xu[1], Josef van Genabith[2], Deyi Xiong[3], and Hongying Zan[1](✉)

[1] Zhengzhou University, Zhengzhou, China
iehyzan@zzu.edu.cn
[2] DFKI and Saarland University, Saarland, Germany
Josef.Van_Genabith@dfki.de
[3] Tianjin University, Tianjin, China
dyxiong@tju.edu.cn

Abstract. Hybrid tabular-textual question answering (QA) requires reasoning from heterogeneous information, and the types of reasoning are mainly divided into numerical reasoning and span extraction. Current numerical reasoning methods simply use LSTMs to autoregressively decode program sequences, and each decoding step produces either an operator or an operand. However, step-by-step decoding suffers from exposure bias, and the accuracy of program generation drops sharply as the decoding steps unfold due to error propagation. In this paper, we propose a non-autoregressive program generation framework, which facilitates program generation in parallel. Our framework, which independently generates complete program tuples containing both operators and operands, can address the error propagation issue while significantly boosting the speed of program generation. Experiments on the ConvFinQA and MultiHiertt datasets show that our non-autoregressive program generation method can bring about substantial improvements over the strong FinQANet (+5.06 Exe Acc and +4.80 Prog Acc points) and MT2Net (+7.97 EM and +6.38 F1 points) baselines, establishing the new state-of-the-art performance, while being much faster (∼21x) in program generation. Finally, with increasing numbers of numerical reasoning steps the performance drop of our method is significantly smaller than that of the baselines.

Keywords: Tabular-Textual Question Answering · Numerical Reasoning · Program Generation

1 Introduction

Most previous QA studies focus on homogeneous data, such as either unstructured text [6,16] or structured data [8,19]. In comparison, hybrid tabular-textual

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 591–603, 2023.
https://doi.org/10.1007/978-3-031-44693-1_46

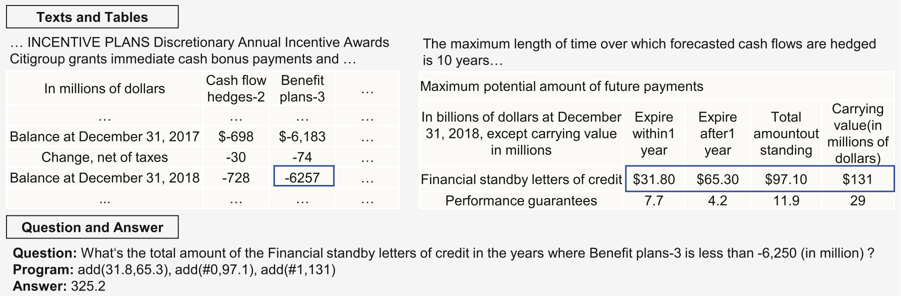
QA [2–4,9,23,24] reasons from heterogeneous information and is more challenging as it often requires numerical reasoning to answer the question in addition to span extraction.

To empower hybrid tabular-textual QA models with numerical reasoning ability, TAGOP [24] uses sequence tagging to extract supporting facts, then performs a single arithmetic operation with one of a number of pre-defined operators. To support multi-step reasoning, FinQANet [3] and MT2Net [23] use an autoregressive LSTM decoder over the RoBERTa [11] representation to gradually generate the program sequence, where each decoding step produces either an operator or an operand. However, this step-by-step autoregressive decoding process suffers from severe exposure bias. During training, the model uses gold references as decoding history (teacher forcing), and learns to rely on the reference decoding history. But the decoding history is very likely to be wrong during inference if the model cannot produce high quality predictions, and wrong predictions in early steps may negatively affect subsequent predictions and lead to further errors in following steps [22]. Unfortunately, this is the case with hybrid QA, where the prediction performance of current methods are far from good, and as a result of error propagation, program generation accuracy drops heavily as the number of decoding steps increases.

In this paper, we propose a **N**on-**A**utoregressive **P**rogram **G**eneration model (NAPG). Instead on generating the program sequence in a step-by-step manner, we only use the encoder representation (without decoding history), and employ an independent numerical reasoning tuple (operator, operands) generator for each reasoning step to predict the operator and its operands. The numerical reasoning tuple generator contains a soft masking mechanism [21] to derive its specific input representation from the encoder representation by highlighting the operand representations of the step, followed by an operator generator, an operand generator and an order predictor. We also utilize a length predictor to control the number of numerical reasoning tuples produced. As the numerical reasoning tuple generator does not leverage any previous decoder history steps, our method prevents the generation with the exposure bias problem and greatly improves the generation speed due to parallelization.

Our main contributions are as follows:

- We propose a non-autoregressive program generation model (NAPG), which can generate the full reasoning programs in parallel. Compared to previous autoregressive generators, our method does not suffer from the exposure bias issue and is much faster due to parallelization.
- In our experiments on the ConvFinQA [4] and MultiHiertt [23] datasets, the NAPG model can bring about substantial improvements over the strong FinQANet (+5.06 Exe Acc and +4.80 Prog Acc points) and MT2Net (+7.97 EM and +6.38 F1 points) baselines, establishing the new state-of-the-art performance, while being ∼21 times as fast in program generation. Our further analysis shows that the performance loss of NAPG is also significantly smaller than the baseline with increasing numbers of numerical reasoning steps.



Texts and Tables

… INCENTIVE PLANS Discretionary Annual Incentive Awards Citigroup grants immediate cash bonus payments and …

In millions of dollars	Cash flow hedges-2	Benefit plans-3	…
…	…	…	…
Balance at December 31, 2017	$-698	$-6,183	…
Change, net of taxes	-30	-74	…
Balance at December 31, 2018	-728	-6257	…
…	…	…	…

The maximum length of time over which forecasted cash flows are hedged is 10 years…

Maximum potential amount of future payments				
In billions of dollars at December 31, 2018, except carrying value in millions	Expire within1 year	Expire after1 year	Total amountout standing	Carrying value(in millions of dollars)
Financial standby letters of credit	$31.80	$65.30	$97.10	$131
Performance guarantees	7.7	4.2	11.9	29

Question and Answer

Question: What‘s the total amount of the Financial standby letters of credit in the years where Benefit plans-3 is less than -6,250 (in million) ?
Program: add(31.8,65.3), add(#0,97.1), add(#1,131)
Answer: 325.2

Fig. 1. An example from the MultiHiertt dataset. In the numerical reasoning question, the system needs to locate which year has less than −6,250 Benefit plans-3 from the first table, and then select the relevant numbers from the second hierarchical table as operands to calculate the answer with addition as the operator. Better viewed in color, the supporting facts are in light blue boxes. (Color figure online)

2 Preliminaries

Task Description. Question answering over hybrid tabular textual data requires reasoning from heterogeneous information, involving numerical reasoning or span extraction. As shown in Fig. 1, given the question Q, the system is to find its answer from tables T and texts E. For some cases, the model only needs to extract an answer span A from the input. For many other cases involving numerical reasoning, the model has to generate a program sequence $G = \{g_0, g_1, ..., g_n\}$, where g_i stands for a token of the program, which is either extracted from the input, or selected from pre-defined special tokens, including operators and special operands, and the probability of an answer A is calculated by summing over the probabilities of all programs G_i from which the answer A can be obtained:

$$P\left(A|T,E,Q\right) = \sum_i P\left(G_i|T,E,Q\right) \tag{1}$$

Fact Retrieving. MT2Net converts data cells of tables into sentences with their row and column headers. As the input text in a document of the MultiHiertt dataset may exceed 3,000 tokens and due to the input length limitation of PLMs, MT2Net first concatenates the question with each sentence as input to train a RoBERTa-based binary-classifier (bi-classifier) for supporting fact classification. Next, it takes the top n sentences based on the supporting fact classification prediction as the input for the next stage. Another classifier is used to determine whether the next stage is span extraction or numerical reasoning.

Span Extraction. MT2Net uses the T5-base model [15] for span extraction questions, where the model takes the concatenation of the question and the sentences containing supporting facts as input, and generates the answer sequence.

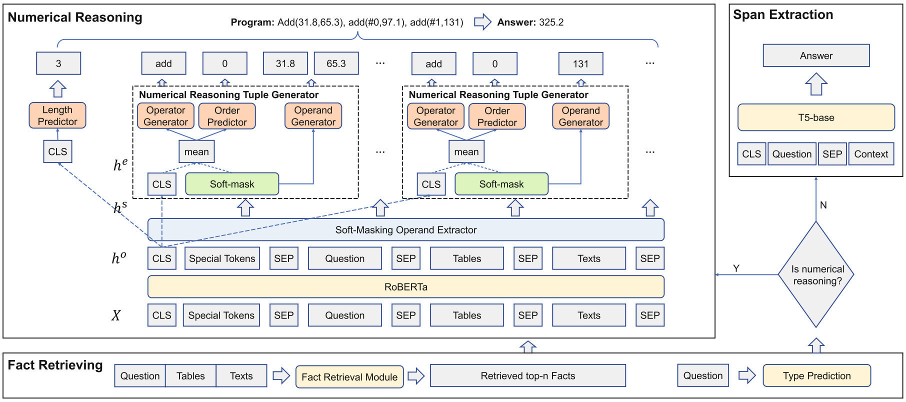


Fig. 2. The NAPG Model.

Autoregressive Numerical Reasoning. Answering the question may require multi-step reasoning. MT2Net first uses RoBERTa as an encoder to obtain the context-aware representations of the question and the sentences containing supporting facts, and concatenates them with the embeddings of pre-defined special tokens, such as the function names, predefined constants, etc. Next, it uses an LSTM decoder to generate the program sequence for the deduction of the answer. Each decoding step makes predictions over the concatenated matrix and selects either an operator or an operand.

3 Our Approach

We present our non-autoregressive program generation model that independently generates the full program sequence to address the exposure bias issue of the step-by-step program generation model and to speed up the generation by supporting better parallelization. The NAPG model framework is shown in Fig. 2. It first uses a bi-classifier to retrieve the most relevant facts, and uses another bi-classifier to identify the question type like MT2Net. For span extraction questions, NAPG uses the same T5-base model as MT2Net to generate the answer given the concatenation of the question and the sentences containing supporting facts. But for numerical reasoning, we design a non-autoregressive approach to program generation, which is quite different from the autoregressive LSTM decoder used by MT2Net.

3.1 Non-Autoregressive Program Generation

We concatenate special tokens (constants within 10 and common order values), question, tables, texts as input to the RoBERTa encoder and it outputs the contextual vectors $\boldsymbol{h}^o$.

Soft-Masking Operand Extractor. We employ a 2-layer feed forward network (FFN) as the Soft-Masking Operand Extractor over the full RoBERTa representation $\boldsymbol{h}^o$ to identify **all operands** of the expected reasoning program in the input. We calculate the probability p^t that the token is an operand. Then we soft mask [21] $\boldsymbol{h}^o$ with p^t.

$$p^t = \text{softmax}\left(\text{FFN}\left(\boldsymbol{h}^o\right)\right) \tag{2}$$

$$\boldsymbol{h}^s = \boldsymbol{h}^o * p^t + \boldsymbol{v}^m * \left(1 - p^t\right) \tag{3}$$

where $\boldsymbol{h}^s$ is the soft-masked representation, $\boldsymbol{v}^m$ stands for the mask embedding, we use a zero vector with all dimensions set to 0 as the mask embedding, and "*" indicates element-wise multiplication.

A large p^t would make the soft masking result close to the original embedding, while a small p^t would turn the result close to the masking embedding. The soft-masking mechanism can thus represent the evidences with a higher priority. Compared to using the classification results for hard masking, soft masking is differentiable and can be trained in an end-to-end manner while alleviating the error propagation issue.

Length Predictor. We employ a multi-class classifier as the length predictor to predict the number of reasoning steps, where each reasoning step contains a complete program tuple (operator, operands, order). The classifier is an FFN layer with the RoBERTa representation of the [CLS] token without soft masking as input.

$$p^{\text{length}} = \text{softmax}\left(\text{FFN}\left([\text{CLS}]\right)\right) \tag{4}$$

Soft-Masking Operand Generator. We also utilize the soft masking mechanism to extract the **two operands** of the specific reasoning step from the input in the Numercical Reasoning Tuple Generator.

$$p^e = \text{softmax}\left(\text{FFN}\left(\boldsymbol{h}^s\right)\right) \tag{5}$$

$$\boldsymbol{h}^e = \boldsymbol{h}^s * p^e + \boldsymbol{v}^m * \left(1 - p^e\right) \tag{6}$$

where $\boldsymbol{h}^e$ is the soft-masked representation for current step. p^e represents the probability that the token is the operand of the current step.

Operator Generator. We define six operators: Addition, Subtraction, Multiplication, Division, Exp, Greater. We average the soft-masked representations $\boldsymbol{h}^e$ of the reasoning step and the embedding of the [CLS] token as input, and use a multi-classifier as the operator generator to select the operator.

$$p^{\text{op}} = \text{softmax}\left(\text{FFN}\left(\text{mean}\left([\text{CLS}]\,|\boldsymbol{h}^e\right)\right)\right) \tag{7}$$

Order Predictor. The order of the two operands matters when the operator is subtraction, division, exp or greater. We also take the mean pooling of the [CLS] token embedding and the soft-masked embeddings $\boldsymbol{h}^e$ as input, and use a bi-classifier to predict the order of the operands (whether their order is as in the input or in the reverse order).

$$p^{\text{order}} = \text{softmax}\left(\text{FFN}\left(\text{mean}\left(\left[\text{CLS}\right]|\boldsymbol{h}^e\right)\right)\right) \tag{8}$$

3.2 Training

To jointly optimize all objectives for numerical reasoning, we minimize the weighted sum of the negative log-likelihood losses of individual modules.

$$\begin{aligned} \mathcal{L} = {} & \lambda^t * \text{NLL}\left(\log\left(p^t\right), r^t\right) + \lambda^{\text{length}} * \text{NLL}\left(\log\left(p^{\text{length}}\right), r^{\text{length}}\right) \\ & + \lambda^e * \sum_{i=0}^{n} \text{NLL}\left(\log\left(p_i^e\right), r_i^e\right) + \lambda^{\text{op}} * \sum_{i=0}^{n} \text{NLL}\left(\log\left(p_i^{\text{op}}\right), r_i^{\text{op}}\right) \\ & + \lambda^{\text{order}} * \sum_{i=0}^{n} \text{NLL}\left(\log\left(p_i^{\text{order}}\right), r_i^{\text{order}}\right) \end{aligned} \tag{9}$$

where NLL stands for the negative log-likelihood loss function, r indicates the ground truths, λ represents the weight of each module, and n is the maximum number of reasoning steps.

3.3 Discussions

The general design of NAPG only involves element-wise computations, a single FFN for length prediction, and 3 sets of FFNs with different parameters but the same architecture for operand generation, operation generation and order prediction respectively. When generating the program tuple sequence, the length predictor only needs to be computed once, and the element-wise computations can be easily parallelized. As for each set of FFNs with different parameters but the same architecture, their activation function can be easily parallelized, and their linear layers with different parameters can be parallelized with the batch matrix-matrix multiplication function implemented in almost all modern linear algebra libraries.

Compared to autoregressive decoding, non-autoregressive counterparts ignore the decoding history which may have a potential negative impact on its coherence, but in case of program generation for numerical reasoning, coherence may be affected less, while **the autoregressive decoding is very likely to be mislead especially for program generation when prediction quality is not high** (<50%, **as shown in Fig.** 3**).**

4 Experiments

4.1 Settings

Dataset. We conducted our experiments on the ConvFinQA[1] and MultiHiertt[2] datasets. The ConvFinQA dataset contains 14,115 data, split into train, development, and test parts with 11,104, 1,490, and 1,521 examples, and it poses a

[1] https://github.com/czyssrs/ConvFinQA.

[2] https://github.com/psunlpgroup/MultiHiertt.

Table 1. Main results on ConvFinQA and MultiHiertt.

	ConvFinQA		MultiHiertt	
	Exe Acc	Prog Acc	EM	F1
GPT-2 (medium)	58.19	57.00	–	–
T5 (large)	58.66	57.05	–	–
TAGOP (RoBERTa-large)	–	–	17.81	19.35
FinQANet (RoBERTa-large)	68.90	68.24	31.72	33.60
MT2Net (RoBERTa-large)	–	–	36.22	38.43
Ours (RoBERTa-base)	69.82	68.84	38.19	38.81
Ours (RoBERTa-large)	**73.96**	**73.04**	**44.19**	**44.81**

great challenge in modeling long-range, complex numerical reasoning paths in real-world conversations [4]. Compared with existing datasets, each document in MultiHiertt contains multiple hierarchical tables and longer unstructured text [23], and it contains 10,440 data items, split into train, development, and test parts with 7,830, 1,044, and 1,566 examples. The test set labels of the ConvFinQA and MultiHiertt datasets are not public.

Evaluation Metrics. We evaluated the performance by Exact Matching (EM) and the adopted numeracy-focused F1 [6] for MultiHiertt, and execution accuracy (Exe Acc) and program accuracy (Prog Acc) for ConvFinQA following previous work.

Baselines. GPT-2 [14] and T5 [15] are two generative models. TAGOP [24] first uses the sequence tagging method to extract facts, then performs only one arithmetic operation with pre-defined operators. FinQANet [3] and MT2Net [23] are able to perform multi-step reasoning, and they both use an autoregressive LSTM decoder to generate the program.

Model Settings. We tuned hyper-parameters on the development set (Sect. 4.4). To fairly compare with existing state-of-the-art results, we adopted all experiment settings of FinQANet and MT2Net, GPT-2 and T5 use medium and large respectively, the rest of baselines are based on the RoBERTa-large model. We train models on a single RTX3090 GPU and set the maximum number of reasoning steps n to 5, 10 for ConvFinQA and MultiHiertt respectively. To focus on program generation and for fair comparison, we only replace the program generation module of MT2Net and FinQANet with NAPG and leave the other parts unchanged. As FinQANet uses a single LSTM to decode either the program sequence or the span to extract according to question type without having an individual span extraction module, we ask the length predictor of NAPG to predict a length of 0 and extract the span with the highest prediction probability directly from the output of the operand extractor in this case to take care of span extraction questions on the ConvFinQA dataset.

4.2 Main Results

We first compare NAPG (with both base and large settings) with our baselines. Results are shown in Table 1.

Table 1 shows that: 1) Pre-trained models (GPT-2 and T5) do not lead to better performance than LSTM, probably due to the fact that they are not pre-trained for the generation of the numerical reasoning program. 2) already with the RoBERTa base setting, our method is able to achieve better performance on both datasets. 3) using the RoBERTa large setting can further boost the performance of NAPG, and lead to large improvements on both ConvFinQA (+5.06/+4.80 Exe/Prog Acc points) and MultiHiertt (+7.97/+6.38 EM/F1 points) datasets.

Table 2. Main results of numerical reasoning on MultiHiertt.

	Dev	
	EM	F1
MT2Net (RoBERTa-large)	41.35	41.35
Ours (RoBERTa-large)	**48.20**	**48.20**

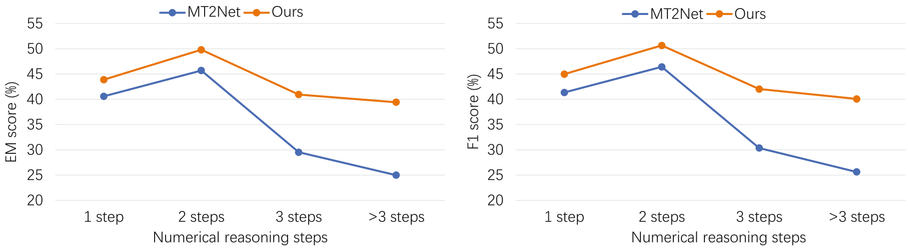


Fig. 3. Performance of different numerical reasoning steps on the development set of MultiHiertt.

As our approach only modifies the program generation part, we also tested the performance of NAPG and MT2Net on all numerical reasoning questions in the development set of MultiHiertt. Results are shown in Table 2.

Table 2 shows that NAPG can lead to large improvements over the MT2Net baseline (+6.85 EM and F1) in numerical reasoning.

4.3 Performance w.r.t. Reasoning Steps

To verify whether NAPG can really address the error accumulation issue of autoregressive program generation, we analyze the performance of NAPG and MT2Net w.r.t. different numbers of reasoning steps. For fairness, we used

Table 3. Effects of different weights of each module.

λ^{t}	$\lambda^{\mathrm{length}}$	λ^{e}	λ^{op}	λ^{order}	base		large	
					EM	F1	EM	F1
1	1	1	1	1	38.60	39.54	44.35	45.29
2					37.84	38.77	**44.92**	**45.86**
	2				37.45	38.39	**44.64**	**45.57**
		2			37.93	38.87	42.72	43.66
			2		**39.27**	**40.21**	43.77	44.71
				2	**39.18**	**40.11**	43.87	44.81
2	1.5				37.55	38.49	45.21	46.15
			2	1.5	**37.93**	**38.87**	**45.79**	**46.72**

Table 4. Time costs for program generation.

Model	Time (s)	Speed-up
LSTM	168.86	1x
Ours	8.04	21x

RoBERTa-large as the encoder of both NAPG and MT2Net. As the test set is not publicly available, our analysis is performed on the development set and the results of MT2Net are from [23]. Results are shown in Fig. 3.

Figure 3 shows that: 1) despite the metrics reporting highest scores with 2 reasoning steps, the general performance trend is descending while increasing the number of reasoning steps. 2) our NAPG approach outperforms the MT2Net in all aspects by a large margin. 3) as the number of reasoning steps increases, the improvements of our method are much larger over the autoregressive MT2Net baseline (+11.41/+11.67 EM/F1 when the number of reasoning steps is 3 and +14.42/+14.43 EM/F1 when it is larger than 3).

The performance drop with increased numbers of reasoning steps with our non-autoregressive method is much smaller than the autoregressive MT2Net, showing the advantage of NAPG in handling questions that require inference with long program sequences. Intuitively, in the generation of longer program sequences, the autoregressive model is more likely to suffer from exposure bias, while the non-autoregressive generation prevents our method from suffering from this issue.

4.4 Ablation Study of Hyper-Parameters

To study the effects of different components in NAPG on the performance, we explored a number of hyper-parameter values for the combination of training losses (Eq. 9) on MultiHiertt. Specifically, we first increase only one of all hyper-parameters to 2 while keeping the others set to 1 in each experiment, and then increase all hyper-parameters that lead to improvements for either the base setting or the large setting together, while assigning the hyper-parameter that leads to more improvements a larger value. Results are shown in Table 3.

Table 3 shows that the best performing settings are different with different model settings. The best setting among all tested cases for the base setting is to use a λ^{op} of 2 while setting the others to 1, and for the large setting is to use a λ^{op} of 2, a λ^{order} of 1.5 while setting the others to 1.

4.5 Program Generation Speed Analysis

Non-autoregressive program generation allows our approach to benefit from parallelization. We compared the program generation speed of NAPG and the LSTM decoder of MT2Net by recording the time costs of the program generation modules on all numerical reasoning questions in the training set of MultiHiertt. Results are shown in Table 4.

Table 4 shows that NAPG is 21 times as fast as MT2Net, showing the substantial advantage of non-autoregressive decoding over autoregressive decoding in terms of speed due to parallelization.

5 Related Work

Hybrid Tabular-Textual QA [2] present the first hybrid tabular-textual QA dataset, HybridQA, by linking table cells to Wiki pages via hyperlinks manually. [3,24] present TAT-QA and FinQA based on financial reports, which require numerical reasoning. TAT-HQA [9] and ConvFinQA [4] are extensions of these two datasets respectively. [23] present the MultiHiertt dataset, which contains multiple hierarchical tables and longer unstructured text.

Numerical Reasoning. Numerical reasoning ability is very important for many NLP tasks [12,18], especially in QA, such as text QA [6,17,20], table QA [8,19], and hybrid tabular-textual QA [3–5,9,10,23,24]. Some works [1,7,13] attempt to inject numerical reasoning ability into pre-trained language models. TAGOP [24] can perform a single arithmetic operation based on predefined operators. FinQANet [3] and MT2Net [23] can perform multi-step reasoning, both of them use the LSTM decoder to autoregressively generate the program.

6 Conclusion

Hybrid tabular-textual question answering (QA) requires reasoning from heterogeneous information, and numerical reasoning is its key challenge compared to extractive QA. To address the severe exposure bias issue of current autoregressive methods when program generation performance is far from good, we present a non-autoregressive program generation (NAPG) framework for numerical reasoning, which facilitates program generation in parallel. Our framework independently generates complete program tuples containing both the operator and its operands. Compared to previous autoregressive decoding methods, NAPG does not suffer from exposure bias, and can significantly boost program generation speed.

Our experiments on the ConvFinQA and MultiHiertt datasets show that: 1) our proposed model can bring about large improvements over the strong FinQANet (+5.06/+4.80 Exe/Prog Acc points) and MT2Net (+7.97/+6.38 EM/F1 points) baselines, establishing the new state-of-the-art performance, while being much faster (∼21x) in program generation. 2) the performance drop of our method is also significantly smaller than the autoregressive LSTM decoder of MT2Net with increasing numbers of numerical reasoning steps.

Acknowledgements. This work is supported the Henan Provincial Department of Science and Technology (232102211041), the Youth Science Foundation Project of the Henan Provincial Natural Science Foundation (232300421386), the National Natural Science Foundation of China (Grant No. 62006211), the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP), and the Zhejiang Lab (Grant No. 2022KH0AB01).

References

1. Berg-Kirkpatrick, T., Spokoyny, D.: An empirical investigation of contextualized number prediction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4754–4764 (2020)
2. Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y.: HybridQA: a dataset of multi-hop question answering over tabular and textual data. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1026–1036 (2020)
3. Chen, Z., et al.: FinQA: a dataset of numerical reasoning over financial data. Proceedings of EMNLP 2021 (2021)
4. Chen, Z., Li, S., Smiley, C., Ma, Z., Shah, S., Wang, W.Y.: ConvFinQA: exploring the chain of numerical reasoning in conversational finance question answering. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 6279–6292. Association for Computational Linguistics. https://aclanthology.org/2022.emnlp-main.421
5. Deng, Y., Lei, W., Zhang, W., Lam, W., Chua, T.S.: PACIFIC: towards proactive conversational question answering over tabular and textual data in finance. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 6970–6984. Association for Computational Linguistics. https://aclanthology.org/2022.emnlp-main.469
6. Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: Drop: a reading comprehension benchmark requiring discrete reasoning over paragraphs. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2368–2378 (2019)

7. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 946–958 (2020)
8. Herzig, J., Nowak, P.K., Mueller, T., Piccinno, F., Eisenschlos, J.: TAPAS: weakly supervised table parsing via pre-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4320–4333 (2020)
9. Li, M., Feng, F., Zhang, H., He, X., Zhu, F., Chua, T.S.: Learning to imagine: integrating counterfactual thinking in neural discrete reasoning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 57–69 (2022)
10. Li, X., Sun, Y., Cheng, G.: TSQA: tabular scenario based question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 13297–13305 (2021)
11. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
12. Pal, K.K., Baral, C.: Investigating numeracy learning ability of a text-to-text transfer model. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3095–3101 (2021)
13. Pi, X., et al.: Reasoning like program executors. arXiv preprint arXiv:2201.11473 (2022)
14. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer (140) (2020)
16. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392 (2016)
17. Ran, Q., Lin, Y., Li, P., Zhou, J., Liu, Z.: NumNet: machine reading comprehension with numerical reasoning. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2474–2484 (2019)
18. Thawani, A., Pujara, J., Ilievski, F., Szekely, P.: Representing numbers in NLP: a survey and a vision. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 644–656 (2021)
19. Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., Paul, S.: TableFormer: robust transformer modeling for table-text encoding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 528–537 (2022)
20. Zhang, Q., et al.: NOAHQA: numerical reasoning with interpretable graph question answering dataset. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 4147–4161 (2021)
21. Zhang, S., Huang, H., Liu, J., Li, H.: Spelling error correction with soft-masked BERT. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 882–890 (2020)
22. Zhang, W., Feng, Y., Meng, F., You, D., Liu, Q.: Bridging the gap between training and inference for neural machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4334–4343, Florence, Italy. Association for Computational Linguistics, July 2019. https://doi.org/10.18653/v1/P19-1426. https://aclanthology.org/P19-1426

23. Zhao, Y., Li, Y., Li, C., Zhang, R.: MultiHiertt: numerical reasoning over multi hierarchical tabular and textual data. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6588–6600 (2022)
24. Zhu, F., et al.: TATQA: a question answering benchmark on a hybrid of tabular and textual content in finance. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3277–3287 (2021)

Mixture-of-Experts for Biomedical Question Answering

Damai Dai[1,2], Wenbin Jiang[2], Jiyuan Zhang[2], Yajuan Lyu[2], Zhifang Sui[1](✉), and Baobao Chang[1]

[1] MOE Key Lab of Computational Linguistics, Peking University, Beijing, China
{daidamai,szf,chbb}@pku.edu.cn
[2] Baidu Inc., Beijing, China
{jiangwenbin,zhangjiyuan01,lvyajuan}@baidu.com

Abstract. Biomedical Question Answering (BQA) has attracted increasing attention in recent years. It is a challenging task because biomedical questions are professional and usually vary widely. Existing question answering methods answer all questions with a homogeneous model, leading to various types of questions competing for the shared parameters, which will confuse the model decision for each single type of question. In this paper, in order to alleviate the parameter competition problem, we propose a Mixture-of-Experts (MoE) based question answering method called MoEBQA that decouples the computation for different types of questions by sparse routing. To be specific, we split a pretrained Transformer model into the bottom and the top blocks. The bottom blocks are shared by all the examples, aiming to capture the general features. The top blocks are extended to an MoE version that consists of a series of independent experts, where each example is assigned to a few experts according to its underlying question type. MoEBQA automatically learns the routing strategy in an end-to-end manner so that each expert tends to deal with a subset of questions it is expert in. We evaluate MoEBQA on three BQA datasets constructed based on real examinations. The results show that within a tolerable computational overhead, our MoE extension significantly improves the accuracy of existing question answering models by 9.4% on average. In addition, we elaborately analyze our MoE modules to reveal how MoEBQA works and find that it can automatically group the questions into human-readable clusters. The code is available at https://github.com/Hunter-DDM/moebqa.

Keywords: Mixture-of-Experts · Biomedical Question Answering

1 Introduction

In recent years, Biomedical Question Answering (BQA) has attracted increasing attention due to its promising application prospect, e.g., supporting the clinical decision for doctors, or being integrated into search engines and chatbots.

D. Dai—Joint work of Peking University and Baidu Inc.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 604–615, 2023.
https://doi.org/10.1007/978-3-031-44693-1_47

Compared with general domain question answering, BQA is more challenging because the biomedical questions are professional and usually vary widely. Existing question answering methods usually answer all questions with a homogeneous model, even if different types of biomedical questions have different focuses and need different problem-solving processes. In this manner, different types of questions will compete for the shared model parameters, which makes a model confused about each single type of question and thus decreases the performance. Therefore, existing question answering methods may not be the best choice for BQA.

In this paper, in order to alleviate the parameter competition problem, we propose a Mixture-of-Experts (MoE) based question answering method called MoEBQA that decouples the computation for different types of questions by sparse routing. To be specific, we split a pretrained Transformer model into the bottom and the top blocks. The bottom blocks are shared by all the examples, aiming to capture the general features among them. By contrast, the top blocks are extended to an MoE version that consists of a series of independent experts and we hope each expert can focus on several types of questions it is expert in. To achieve this, we assign each example to a few experts according to its question representation, which implies its underlying question type. During training, MoEBQA automatically learns the routing strategy in an end-to-end manner so that the question representations will be grouped into clusters and each expert tends to learn how to deal with a specific subset of questions it needs to be expert in. With the MoE extension, we increase the upper bound of the model performance since each type of question is only handled by the proper set of experts and the parameter competition problem is alleviated.

We evaluate MoEBQA on three BQA datasets constructed based on real examinations, including MedQA [10], HEAD-QA [25], and NLPEC [14]. Experimental results show that MoEBQA significantly improves the accuracy of existing models by 9.4% on average within a tolerable computational overhead. In addition, we elaborately analyze our MoE modules to validate our design and reveal how MoEBQA groups the questions into human-readable clusters.

2 Multiple-Choice Question Answering

In this section, we formulate the multiple-choice question answering task. In addition, we introduce a typical paradigm for this task that leverages pretrained Transformer models such as BERT [5].

Given a context C, a question Q, and a set of candidate options $\mathcal{O} = \{O_1, O_2, ..., O_n\}$, the multiple-choice question answering task requires a model to select the correct option O_a from the candidate set $\mathcal{O}$. In the typical paradigm, for each option, we first jointly encode the context C, the question Q, and the option O_i to obtain an overall representation. Taking BERT as an example encoder, we concatenate C, Q, O_i, and the special tokens for BERT to form the input sequence $I =$ "[CLS] C Q [SEP] O_i [SEP]" and feed it into BERT to obtain the overall representation $\mathbf{p}_i$:

$$H = \mathrm{BERT}(I), \quad \mathbf{p}_i = \mathrm{Pooling}(H), \tag{1}$$

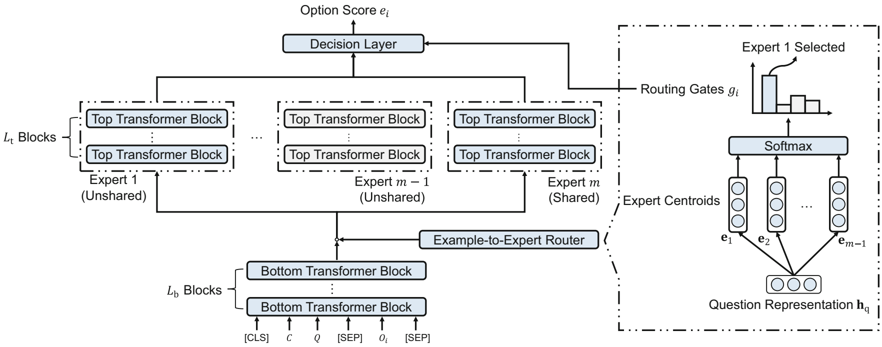


Fig. 1. Illustration of the architecture of MoEBQA. The shared bottom blocks capture general features among all the examples. The top blocks are extended to an MoE version that consists of a shared expert and m-1 unshared experts.

where $H \in \mathbb{R}^{d \times len}$ is the hidden states computed by BERT, and Pooling$(\cdot)$ pools the hidden states into a single vector $\mathbf{p}_i \in \mathbb{R}^d$ (e.g., taking the hidden state of the [CLS] token as $\mathbf{p}_i$). After this, we project each $\mathbf{p}_i$ into a scalar score e_i, and determine the predicted answer as follows:

$$e_i = \mathbf{q}^T \mathbf{p}_i, \quad P(O_i|C,Q) = \alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}, \quad \hat{a} = \arg\max_i \alpha_i, \tag{2}$$

where $\mathbf{q} \in \mathbb{R}^d$ is a trainable vector, e_i is the matching score for the i-th option, the scalar α_i is the predicted probability that the i-th option is the answer, and $\hat{a}$ is the index of the predicted answer.

The training object is to minimize the negative log-likelihood loss: $\mathcal{L}_{\text{task}} = -\sum_{C,Q,\mathcal{O}} \log P(O_a|C,Q)$, where O_a is the ground truth option.

3 Method: MoEBQA

Under the paradigm described in Sect. 2, MoEBQA extends the Transformer to an MoE version. Although our extension is applicable for all the pretrained Transformers, we take BERT [5] as an example backbone for simplicity.

3.1 Overview of MoEBQA

As illustrated in Fig. 1, we split a pretrained Transformer into two parts: L_b bottom Transformers blocks and L_t top Transformers blocks. We keep the bottom blocks shared by all examples, but extend the top blocks to an MoE version that consists of m experts, including a shared one and m-1 unshared ones. At the MoE modules, a router assigns each example to the shared expert and an unshared expert according to its question representation. In this manner, each unshared expert tends to answer a subset of questions it is expert in, which alleviates the parameter competition among different types of questions.

3.2 Shared Bottom Blocks

Although there exists the parameter competition among different types of questions, the examples still share some general features (e.g., general shallow text patterns in the biomedical domain). Therefore, we keep the bottom L_b Transformer blocks shared by all the examples to capture the general features and preliminarily understand the texts.

For each input sequence I with *len* tokens, the bottom blocks of our model encode it into a series of hidden states $H' \in \mathbb{R}^{d\times len}$:

$$H' = \text{BERT}_{\texttt{bottom}}(I) = [\mathbf{h}_1^{L_\text{b}}; \mathbf{h}_2^{L_\text{b}}; ...; \mathbf{h}_T^{L_\text{b}}], \tag{3}$$

where $\text{BERT}_{\texttt{bottom}}$ denotes the bottom L_b Transformer blocks of BERT, and $\mathbf{h}_t^{L_\text{b}} \in \mathbb{R}^d$ is the hidden state of the t-th token after the bottom L_b blocks.

3.3 MoE-Extended Top Blocks

We extend the top L_t blocks to an MoE version. To be specific, we copy the top blocks for m times to produce a set of experts $\{\text{BERT}_{\texttt{top}}^1, \text{BERT}_{\texttt{top}}^2, ..., \text{BERT}_{\texttt{top}}^m\}$, where $\text{BERT}_{\texttt{top}}^m$ is shared and the others are not. For each example, we assign it to two experts, including the shared expert and an unshared expert.

In order to select the unshared expert for an example, we first compute the affinities between its question and the unshared experts. Let q_s and q_e be the start index and the end index of the question in the input sequence I, respectively. We first compute the question representation $\mathbf{h}_\text{q} \in \mathbb{R}^d$ by mean pooling the token representations of the question in H':

$$\mathbf{h}_\text{q} = \text{MeanPooling}(\mathbf{h}_{q_\text{s}:q_\text{e}}^{L_\text{b}}), \tag{4}$$

where $\mathbf{h}_{q_\text{s}:q_\text{e}}^{L_\text{b}} \in \mathbb{R}^{d\times(q_\text{e}-q_\text{s}+1)}$ denotes a set of continuous vectors whose indices range from q_s to q_e. For each unshared expert $\text{BERT}_{\texttt{top}}^i$, we define a trainable centroid vector $\mathbf{e}_i \in \mathbb{R}^d$ for it. Then, we compute a scalar s_i as the affinity between the question and the i-th expert:

$$s_i = \mathbf{e}_i^\top \mathbf{h}_q. \tag{5}$$

Then, given the affinity score s_i, we greedily select the t-th expert with the highest affinity as the target expert that the example will be assigned to:

$$t = \arg\max_i s_i. \tag{6}$$

Finally, we compute the output of the MoE-extended top blocks as

$$g_i = \frac{\exp(s_i)}{\sum_{j=1}^{m-1}\exp(s_j)}, \tag{7}$$

$$H = (1-g_t)\,\text{BERT}_{\texttt{top}}^m(H') + g_t\,\text{BERT}_{\texttt{top}}^t(H'), \tag{8}$$

Table 1. Official data splits of three biomedical question answering datasets.

Datasets	Training	Validation	Test	Total
MedQA	10,178	1,272	1,273	12,723
HEAD-QA	2,657	1,366	2,742	6,765
NLPEC	18,703	2,500	547	21,750

where the scalar g_t is a softmax gate that controls how much the t-th unshared expert will be used. Considering the gate g_t, if the selected unshared expert $\text{BERT}^t_{\text{top}}$ is beneficial for answering the question, optimizing the training objective $\mathcal{L}_{\text{task}}$ will urge the gate to be larger; otherwise, the model will tend to use the expert sparely by suppressing the gate. Therefore, the gate g_t can encourage similar questions to be assigned to the same unshared expert that is beneficial to them, i.e., automatically grouping the examples into clusters according to their underlying question types.

Balance Loss. Ideally, we expect each expert to have a relatively balanced example load to guarantee parameter utilization. Otherwise, some experts will degrade into ineffective ones since they are seldom activated. Therefore, inspired by [6, 12], we use a differentiable balance loss to avoid imbalanced expert loads. Let c_i denote the number of examples that the i-th expert has been assigned in the training history, and the balance loss $\mathcal{L}_{\text{bal}}$ is computed as follows:

$$\mathcal{L}_{\text{bal}} = (m-1) \sum_{i=1}^{m-1} \frac{c_i}{\sum_{j=1}^{m-1} c_j} g_i, \tag{9}$$

Intuitively, if an expert is overloaded, the weight $\frac{c_i}{\sum_{j=1}^{m-1} c_j}$ will be higher than the average, and thus the balance loss tends to decrease the affinities related to the expert to drop some examples. Otherwise, if an expert is relatively unoccupied, the balance loss will increase its affinities to capture more examples. The balance loss is minimized when the experts have absolutely balanced loads. With the balance loss, the final loss to optimize is:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{task}} + \beta \mathcal{L}_{\text{bal}}, \tag{10}$$

where the balance factor β is a hyper-parameter.

4 Experiments

4.1 Datasets

We evaluate MoEBQA on three BQA datasets, including MedQA [10], HEAD-QA [25], and NLPEC [14]. **MedQA** is extracted from the National Medical Board Examinations in the USA, Mainland China, and Taiwan. In this paper, we use its English subset, which contains 12, 723 examples in total. Each example in the dataset contains a question and several options with the correct one annotated. **HEAD-QA** is created from real examinations spanning from 2013 to 2017 that are organized by the Spanish government. HEAD-QA contains 6, 765 examples in total, where each example has a question and several options with the correct one annotated. **NLPEC** is a Chinese dataset, constructed based on the National Licensed Pharmacist Examination in China. It contains 21, 750 examples in total, where each example in NLPEC has a question, several options with the correct one annotated, and relevant evidence extracted from the official exam guidebook. The official data splits of these datasets are shown in Table 1.

4.2 Experimental Setup

All experiments are conducted on NVIDIA V100 GPUs with 32 GB memory. We evaluate the model on the validation set after each epoch and use the best checkpoint to obtain the test accuracy. Due to the space limit, in the following, we only describe some key settings in our final experiments for each dataset.

For MedQA, we use PubMedBERT [7] as the pretrained backbone model, which shares the same architecture as BERT-base [5], but is pretrained from scratch on biomedical domain corpora. We use AdamW [17] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as the optimizer, and set the learning rate to 3e−5. We set the batch size to 16 and train the model for 5 epochs. We set L_b to 10 and L_t to 2. The number of experts is set to 5 and the balance factor β is set to 0.01. **For HEAD-QA**, we also use PubMedBERT as the backbone model. Since the questions in HEAD-QA do not contain enough contextual information, we follow [25] to use the DrQA document retriever [3] to retrieve a relevant document from Wikipedia as the context. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as the optimizer, and set the learning rate to 5e−5. We set the batch size to 8 and train the model for 2 epochs. We set L_b to 10 and L_t to 2. The number of experts is set to 3 and the balance factor β is set to 0.003. **For NLPEC**, We use RoBERTa-large [16] as the backbone model for NLPEC. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as the optimizer, and set the learning rate to 2e−5. We set the batch size to 16 and train the model for 35 epochs. We set L_b to 20 and L_t to 4. The number of experts is set to 5 and the balance factor β is set to 0.001.

4.3 Results

On **MedQA**, we compare MoEBQA with ClinicalBERT [1], BioBERT [11], BERT [5], BioRoBERTa [8], RoBERTa [16], and PubMedBERT [7] + MoP [18]. In Table 2, 3, and 5, except for our result, the other results are reported in their original or reproduced papers. From Table 2, we find that MoEBQA improves the test accuracy of PubMedBERT by 18.5%. On **HEAD-QA**, we compare MoEBQA with BiDAF [21], TFIDF-IR [25], IR + BERT [15], IR + BioBERT [15], Multi-step Reasoner [15], PubMedBERT [7], and MurKe [15]. In Table 3, we find that MoEBQA improves the test accuracy of PubMedBERT 2.4% and achieves the same performance as MurKe. Note that MurKe needs 30 supporting documents for reference while PubMedBERT + MoEBQA needs only one supporting document to achieve the same accuracy. Therefore, from

Table 2. Accuracy on MedQA.

Methods	Valid	Test
ClinicalBERT-base	33.7	32.4
BioBERT-base	34.3	34.1
BERT-base	33.9	34.3
BioRoBERTa-base	35.1	36.1
RoBERTa-large	35.2	35.0
BioBERT-large	36.1	36.7
PubMedBERT	–	35.1
PubMedBERT + MoP	–	38.0
PubMedBERT + MoEBQA	**39.9**	**41.6**

Table 3. Accuracy on HEAD-QA.

Methods	Valid	Test
BiDAF	–	30.3
TFIDF-IR	–	37.2
IR + BERT	–	35.0
IR + BioBERT	–	36.4
Multi-step Reasoner	–	42.9
PubMedBERT	43.5	45.6
PubMedBERT + MoEBQA	**44.3**	**46.7**
MurKe (30 docs)	–	46.7
MurKe (10 docs)	–	<40.0

Table 4. Number of examples that a model can process per second.

Datasets	Training		Inference	
	Dense Backbone	+MoEBQA	Dense Backbone	+MoEBQA
MedQA	39.3	30.0	113.3	91.3
HEAD-QA	15.5	12.8	60.5	50.3
NLPEC	6.4	5.5	19.2	16.5

the aspect of the modeling ability, MoEBQA is stronger than all the previous methods. On **NLPEC**, we use BiDAF, BERT, RoBERTa, ERNIE [23], and KMQA [14] as baselines. From Table 5, we observe that MoEBQA improves the test accuracy of RoBERTa-large by 7.4%.

In order to show the computational overhead introduced by our MoE extension, in Table 4, we report the number of examples that the dense backbone model and MoEBQA can process per second. Compared with the dense model, MoEBQA is slightly slower (about 15%–20%) since we calculate two experts in the top blocks. However, considering the significant performance improvement, the computational overhead introduced by our MoE extension is tolerable.

5 Analysis and Discussion

5.1 Ablation Studies

We show the ablation studies of MoEBQA based on the MedQA dataset in Table 6 to validate the effectiveness of each component of our method. Firstly, if we remove the balance loss, unbalanced expert loads will affect the parameter utilization and thus decrease the test accuracy by 1.5. Secondly, if we replace the gate g_t (see Eq. (7)) with a constant 0.5, no useful signal will be propagated from the task loss back to the routing strategy. As a result, the model cannot learn how to group questions and will produce worse question clusters where similar questions cannot be grouped together, leading to a decrease of 1.9 in the test accuracy. Thirdly, if we remove the shared expert and all the experts are unshared, the test accuracy will drop by 1.9. This result suggests

Table 5. Accuracy on NLPEC.

Methods	Valid	Test
BiDAF	52.7	43.6
BERT-base	64.2	52.2
ERNIE	64.7	53.4
RoBERTa-large	70.8	57.9
BERT-base + KMQA	67.9	57.1
RoBERTa-large + KMQA	71.1	61.8
RoBERTa-large + MoEBQA	**72.8**	**62.2**

Table 6. Ablation studies of MoEBQA.

Methods	Valid	Test
MoEBQA	39.9	41.6
w/o Balance Loss	38.6 (1.3 ↓)	40.1 (1.5 ↓)
w/o Gate Signal	39.2 (0.7 ↓)	39.7 (1.9 ↓)
w/o Shared Expert	38.0 (1.9 ↓)	39.7 (1.9 ↓)
w/o MoE Extension	38.3 (1.6 ↓)	38.7 (2.9 ↓)
Routing by Question	39.9	41.6
Routing by [CLS]	37.6 (2.3 ↓)	38.2 (3.4 ↓)

that with the same number of parameters, a shared expert that improves the parameter efficiency is also indispensable. Finally, if we disable the whole MoE extension, the test accuracy will significantly drop by 2.9, which again validates the effectiveness of MoEBQA.

In addition, we also investigate whether the question representation is a suitable routing feature for MoEBQA. In this paper, we use the question representation as the routing feature. If we use the whole example representation (i.e., the representation of the [CLS] token) for routing, the test accuracy will drop by 3.4. This suggests that the model can learn a better routing strategy according to the question representation instead of the whole example representation. It also supports our conclusion that MoEBQA works by automatically grouping the questions into clusters according to their underlying question types.

5.2 Investigation of MoE Architectures

Based on the MedQA dataset, we investigate the performance of different MoE architectures, including different numbers of unshared experts and different splitting ratios between the bottom and top Transformers blocks.

The performance of different numbers of unshared experts is plotted in Fig. 2(a). We find that adding experts can improve the performance when the number of unshared experts is smaller than 4. However, too many experts will harm the performance instead, since the number of experts affects the granularity of the question clusters. If the number of experts is too large, each question cluster will be too small, which will damage the generalization of the model.

The performance of different splitting ratios between L_b and L_t is plotted in Fig. 2(b). When L_t is 0, the model is equivalent to a dense model that does not have MoE modules. Based on it, extending 2 top blocks to the MoE version can significantly boost the performance. However, continuing to increase L_t will lead to poorer performance. The results prove that the shared bottom blocks in MoEBQA, which capture the general features among all the examples, are also indispensable and we should allocate a proper proportion to L_b.

5.3 Effects of Balance Factor

Figure 3 shows the effects of different balance factors on the MedQA dataset. When the balance factor is 0, the balance loss does not take effect. As we grad-

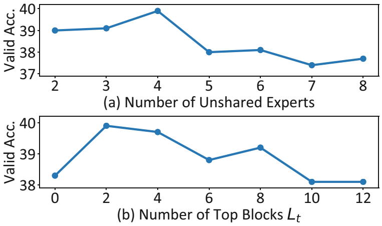


Fig. 2. Performance of different numbers of unshared experts (a) and top blocks (b).

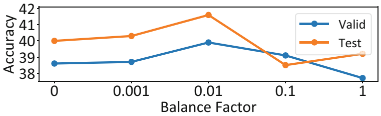


Fig. 3. Effects of balance factors.

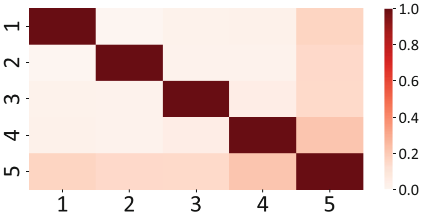


Fig. 4. Normalized similarities between experts.

ually increase the balance factor, we have the following observations. Within a tolerable range (0 → 0.01), a larger balance factor can improve the performance by making the expert loads more balanced, and we achieve the best performance when the balance factor is 0.01. However, if the balance factor is too large (0.01 → 1), the balance loss will overwhelm the primary loss and thus damage the performance. Therefore, the performance with the balance factor of 1 is even worse than that when not applying the balance loss.

5.4 Diversity of Experts

In order to demonstrate the diversity of different experts, we show the similarity between each pair of experts in Fig. 4. To be specific, for the best model trained on MedQA, we flatten all the parameters in each expert, calculate the cosine similarity between each pair of experts, and normalize the similarities into the range of $[0, 1]$. From the figure, we observe that each expert has the highest similarity with itself and a moderate similarity with the shared expert (Expert 5). Meanwhile, each unshared expert has relatively low similarities with the other unshared experts. These results prove that the unshared experts have considerable diversity and the shared expert tends to understand common information.

5.5 Case Study

We present a case study of MoEBQA in Table 7 to reveal the behavior of our MoE modules intuitively. For each expert, we compute its affinities with all the

Table 7. Three representative questions with the highest affinities to each expert.

Expert	Questions with Highest Affinities	Topic
1	What is the most likely organism responsible? Which of the following is the most likely pathogen? Which of the following is the most probable cause of his complaints?	Pathogenesis
2	Which of the following is the best next step in management? Which of the following is the most appropriate next step in management? Which of the following is the initial treatment of choice for the patient?	Treatments
3	Further evaluation is most likely to show which of the following? This drug is most likely to result in which of the following? Microscopic examination of the mass will most likely show which of the following?	Deduction
4	What disease is the child suffering from? What vitamin deficiency is this woman most likely suffering from? What is the pathophysiology of this patient's condition?	Diagnosis

questions in the MedQA validation set. Then, we present three representative questions among the top-5 questions with the highest affinities. From Table 7, we observe that the questions assigned to the same expert tend to express a common topic. For example, (1) the questions assigned to Expert 1 require identifying the pathogenesis, (2) the questions assigned to Expert 2 ask for suggestions on the treatments, (3) the questions assigned to Expert 3 need to deduce an expected result, (4) and the questions assigned to Expert 4 aim to diagnose the disease. These cases prove that MoEBQA can automatically group the questions into clusters according to their underlying question types. By dealing with each cluster of questions separately, MoEBQA achieves better performance.

6 Related Work

Biomedical Question Answering. In recent years, many BQA datasets are constructed [10,14,19,24,25]. Among them, MedQA [10], HEAD-QA [25], and NLPEC [14] are constructed based on medical examinations in different countries and districts. Existing BQA methods are usually based on biomedical domain pretrained language models [1,7,8,11].

Mixture of Experts. [9] first propose the MoE technique to compute different examples with independent experts. [22] first apply MoE to build large-scale LSTM language models. Recently, MoE is applied on Transformers. GShard [12] and Switch Transformer [6] dynamically learn to route each input token to one or several experts with a balance loss. Without an explicit balance loss, BASE Layer [13] uses the auction algorithm [2] to directly find a balanced assignment. Hash Layer [20] uses a pre-designed token-level hash table as the routing strategy. StableMoE [4] first learns to route and then fixes the learned routing strategy for more stable training.

7 Conclusions

We point out the parameter competition problem that limits the BQA performance and propose an MoE-based method MoEBQA. It automatically groups questions into clusters and decouples the computation for different types of questions by sparse routing. We evaluate MoEBQA on three BQA datasets and find our MoE extension can significantly improve the accuracy of existing models. In addition, we validate the effectiveness of each component and reveal the working mechanism of MoEBQA. Note that the idea of the MoE extension is not limited to BQA, so it has the potential to be generalized to other tasks in the future.

Acknowledgement. Damai Dai, Zhifang Sui, and Baobao Chang are supported by the National Key Research and Development Program of China 2020AAA0106700 and NSFC project U19A2065.

References

1. Alsentzer, E., et al.: Publicly available clinical BERT embeddings. CoRR abs/1904.03323 (2019). http://arxiv.org/abs/1904.03323
2. Bertsekas, D.P.: Auction algorithms for network flow problems: a tutorial introduction. Comput. Optim. Appl. **1**(1), 7–66 (1992). https://doi.org/10.1007/BF00247653
3. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: ACL 2017, pp. 1870–1879 (2017). https://doi.org/10.18653/v1/P17-1171
4. Dai, D., Dong, L., Ma, S., Zheng, B., Sui, Z., Chang, B., Wei, F.: StableMoE: stable routing strategy for mixture of experts. In: ACL 2022 (2022)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/n19-1423
6. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. CoRR abs/2101.03961 (2021). https://arxiv.org/abs/2101.03961
7. Gu, Y., et al.: Domain-specific language model pretraining for biomedical natural language processing. CoRR abs/2007.15779 (2020). https://arxiv.org/abs/2007.15779

8. Gururangan, S., et al.: Don't stop pretraining: adapt language models to domains and tasks. In: ACL 2020, pp. 8342–8360 (2020). https://doi.org/10.18653/v1/2020.acl-main.740
9. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79
10. Jin, D., Pan, E., Oufattole, N., Weng, W., Fang, H., Szolovits, P.: What disease does this patient have? A large-scale open domain question answering dataset from medical exams. CoRR abs/2009.13081 (2020). https://arxiv.org/abs/2009.13081
11. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinform. **36**(4), 1234–1240 (2020). https://doi.org/10.1093/bioinformatics/btz682
12. Lepikhin, D., et al.: GShard: scaling giant models with conditional computation and automatic sharding. In: ICLR 2021 (2021). https://openreview.net/forum?id=qrwe7XHTmYb
13. Lewis, M., Bhosale, S., Dettmers, T., Goyal, N., Zettlemoyer, L.: BASE layers: simplifying training of large, sparse models. In: ICML 2021. Proceedings of Machine Learning Research, vol. 139, pp. 6265–6274 (2021). http://proceedings.mlr.press/v139/lewis21a.html
14. Li, D., Hu, B., Chen, Q., Peng, W., Wang, A.: Towards medical machine reading comprehension with structural knowledge and plain text. In: EMNLP 2020, pp. 1427–1438 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.111
15. Liu, Y., Chowdhury, S., Zhang, C., Caragea, C., Yu, P.S.: Interpretable multi-step reasoning with knowledge extraction on complex healthcare question answering. CoRR abs/2008.02434 (2020). https://arxiv.org/abs/2008.02434
16. Liu, Yet al.: RoBERTa: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692 (2019). http://arxiv.org/abs/1907.11692
17. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in Adam. CoRR abs/1711.05101 (2017). http://arxiv.org/abs/1711.05101
18. Meng, Z., Liu, F., Clark, T.H., Shareghi, E., Collier, N.: Mixture-of-partitions: infusing large biomedical knowledge graphs into BERT. In: EMNLP 2021, pp. 4672–4681 (2021). https://aclanthology.org/2021.emnlp-main.383
19. Pampari, A., Raghavan, P., Liang, J.J., Peng, J.: emrQA: a large corpus for question answering on electronic medical records. In: EMNLP 2018, pp. 2357–2368 (2018). https://doi.org/10.18653/v1/d18-1258
20. Roller, S., Sukhbaatar, S., Szlam, A., Weston, J.: Hash layers for large sparse models. CoRR abs/2106.04426 (2021). https://arxiv.org/abs/2106.04426
21. Seo, M.J., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: ICLR 2017 (2017). https://openreview.net/forum?id=HJ0UKP9ge
22. Shazeer, N., et al.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In: ICLR 2017 (2017). https://openreview.net/forum?id=B1ckMDqlg
23. Sun, Y., et al.: ERNIE: enhanced representation through knowledge integration. CoRR abs/1904.09223 (2019). http://arxiv.org/abs/1904.09223
24. Tsatsaronis, G., et al.: BioASQ: a challenge on large-scale biomedical semantic indexing and question answering. In: AAAI 2012 (2012). http://www.aaai.org/ocs/index.php/FSS/FSS12/paper/view/5600
25. Vilares, D., Gómez-Rodríguez, C.: HEAD-QA: a healthcare dataset for complex reasoning. In: ACL 2019, pp. 960–966 (2019). https://doi.org/10.18653/v1/p19-1092

Fine-Grained Question-Answer Matching via Sentence-Aware Contrastive Self-supervised Transfer

Jingjing Wang, Jiamin Luo, and Guodong Zhou(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
{djingwang,gdzhou}@suda.edu.cn, 20204027003@stu.suda.edu.cn

Abstract. Previous studies always consider the question-answer (QA) matching task as a one-to-one text matching problem. This study builds upon existing research by expanding the scope to a many-to-many mapping scenario and proposes a new fine-grained QA matching (FQAM) task, which aims to accurately predict the many-to-many matching relationship between all sub-questions and sub-answers within each QA pair. Particularly, a meticulously annotated corpus of high quality is constructed specifically for FQAM to facilitate this research. On this basis, owing to the challenge of expensive data annotation associated with FQAM, we propose a sentence-aware contrastive self-supervised transfer (SCST) approach to transfer the sentence alignment information pre-trained with massive unannotated QA pairs to assist in FQAM. Experimental evaluations conducted on our annotated corpus demonstrate the importance of utilizing sentence alignment information from unannotated QA pairs in FQAM and justify the effectiveness of our approach in capturing such information.

Keywords: Fine-grained Question-Answer Matching · Sentence Alignment Information · Contrastive Self-supervised Learning

1 Introduction

Question-answer (QA) matching plays a crucial role in the fields of NLP and has garnered significant attention due to its diverse applications spanning from fundamental technology services to business intelligence, such as reading comprehension [1,2] and intelligent agent [3,4], which focuses on predicting the one-to-one relationship (*matching* or *non-matching*) between a given QA text pair.

In real-life communications, it is worth noting that individuals often pose multiple (not just one) questions simultaneously, while answer providers are accustomed to answering each of these questions individually. Especially, such phenomenon (e.g., the QA pair with the many-to-many style shown in Fig. 1) is rather pervasive in e-commerce platforms, e.g., *Amazon*, since potential customers tend to ask multiple questions w.r.t. the different aspects of products

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 616–628, 2023.
https://doi.org/10.1007/978-3-031-44693-1_48

A QA Pair with the Many-to-many Style

Q: Is this phone running quickly enough? [Q1] How about the battery capacity? [Q2]

A: The battery capacity has grown by 33%. [A1] But, the speed is disappointing. [A2] I still enjoy the appearance anyway. [A3]

Fine-grained QA Matching

- Input: A QA Text Pair

- Output: $\text{Q1} \xrightarrow{✗} \text{A1}$; $\text{Q1} \xrightarrow{✓} \text{A2}$; $\text{Q1} \xrightarrow{✗} \text{A3}$;
$\text{Q2} \xrightarrow{✓} \text{A1}$; $\text{Q2} \xrightarrow{✗} \text{A2}$; $\text{Q2} \xrightarrow{✗} \text{A3}$.

Fig. 1. An example for illustrating the proposed fined-grained QA matching (FQAM) task, where ✓ and ✗ denote *matching* and *non-matching* respectively.

before making the purchase decisions. Unfortunately, all sub-questions and sub-answers of a single-turn QA are usually integrated into a single question text and a single answer text respectively, thereby making the traditional one-to-one QA matching inapplicable. To this end, we propose a new fine-grained QA matching (FQAM) task to recognize the many-to-many mapping relationship between all sub-questions and sub-answers within each QA text pair.

Unlike the traditional one-to-one QA matching virtually not requiring manual annotation, the data annotation for FQAM is rather expensive since the many-to-many relationship in each QA pair needs to be manually annotated sentence by sentence. Thus, the data scale of FQAM is limited, inevitably restricting the performance. Inspired by recent pre-trained language models (e.g., BERT [5]) having achieved SOTA performances in many text matching tasks, a potential solution to remedy the above issue is to further train the pre-trained models (e.g., BERT) with the large-scale and easy-obtained unannotated QA pairs, and then fine-tune[1] BERT to help FQAM. Despite this, we believe this solution still fails to control two kinds of key information flows for FQAM, i.e., *sentence alignment* and *contrastive noise* information.

For one thing, prior pre-trained models (e.g., BERT) consistently neglect to explicitly capture the *sentence alignment* information between QA, while this information is rather crucial for FQAM. Take Fig. 1 as an example, sub-question **Q1** is exactly located in the first sentence of the question and the aligned sub-answer **A2** is exactly located in the second sentence of answer. Inspired by this, self-supervisedly learning *sentence alignment* information with massive unannotated QA pairs may powerfully contribute to aligning sub-questions with sub-answers. Thus, a better-behaved pre-trained model for FQAM should incorporate the sentence alignment information between QA during pre-training with unannotated QA pairs.

For another, to learn the sentence alignment information between QA, a feasible way is to mask the sentence and then recover it with some pretext tasks

[1] Except for further pre-training BERT, we can also use these unannotated QA pairs to perform a simple one-to-one QA matching task for fine-tuning BERT, i.e., the baselines with "+ QA Pair" shown in Table 2.

like word and span masking [5,6]. However, when processing the text matching tasks, BERT and its variants always leverage a mark "`[SEP]`" to concatenate the QA pair into a sequence for implicitly learning the QA matching information. This is suboptimal for FQAM since the sequence auto-regressive property may bring much noise from the question when we recover the masked sentence in the answer, and vice versa. Take Fig. 1 as an example, when we mask sub-answer **A2**, only sub-question **Q1** contributes to recovering **A2**. In contrast, the nearer sub-question **Q2** is noisy and should be filtered as much as possible. In this study, this noisy sub-question **Q2** is defined as the *contrastive noise* for sub-answer **A2**. Obviously, treating QA as a sequence cannot filter this noise well. We believe a better-behaved pre-trained model for FQAM should treat the QA pair as two parallel units and explicitly filter this noise.

In this paper, we propose a new two-step self-supervised learning framework, namely Sentence-aware Contrastive Self-supervised Transfer (SCST) approach, to tackle the above two challenges. In the first step, we propose a sentence-aware contrastive self-supervised learning model for pre-training the massive unannotated QA pairs with two pretext tasks (i.e., sentence retrieval and generation). Wherein, a novel contrastive bidirectional attention encoder is designed to capture the *sentence alignment* information between QA and meantime filter the aforementioned *contrastive noise*. In the second step, we propose a sequence decoding model to perform FQAM, and transfer the fine-tuned parameters of the above pre-trained contrastive bidirectional attention encoder to initialize the QA pair encoder inside this sequence decoding model for performing FQAM. Experimental results demonstrate the impressive effectiveness of the SCST approach to FQAM over the SOTA baselines.

2 Approach

In this section, we introduce our SCST approach, the framework of which is shown in Fig. 2. Similar to the prior self-supervised frameworks [14], SCST also consists of two steps. The first step is a sentence-aware contrastive self-supervised learning model for pre-training the massive unannotated QA pairs. The second step is a sequence decoding model for performing FQAM, aiming to transfer the pre-trained knowledge of the first step to boost the FQAM performance.

2.1 Step1: Sentence-Aware Contrastive Self-supervised Learning

Given an unannotated QA text pair $[\mathbf{Q}, \mathbf{A}]$, we first adopt sentence segmentation tool[2] to segment question and answer texts into sentence sequences $\{\mathbf{q}_1, ..., \mathbf{q}_m\}$ and $\{\mathbf{a}_1, ..., \mathbf{a}_n\}$ respectively. Then, we leverage a mark "[SEN]" to distinguish and concatenate the sentences of each question or answer in the following way:

$$\begin{array}{llllllll} \mathbf{Q:} & \mathbf{q}_1 & [\mathtt{SEN}] & \mathbf{q}_2 & [\mathtt{SEN}]\ldots & \mathbf{q}_m & [\mathtt{SEN}] \\ \mathbf{A:} & \mathbf{a}_1 & [\mathtt{SEN}] & \mathbf{a}_2 & [\mathtt{SEN}]\ldots & \mathbf{a}_n & [\mathtt{SEN}] \end{array} \tag{1}$$

[2] http://stanfordnlp.github.io/CoreNLP.

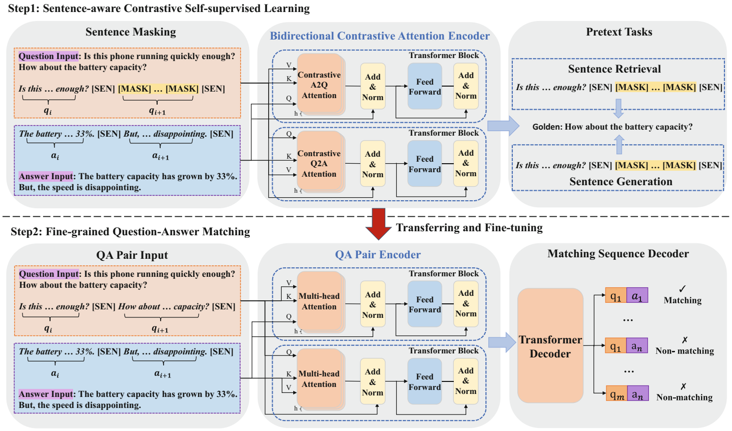

Step1: Sentence-aware Contrastive Self-supervised Learning
Sentence Masking
Question Input: Is this phone running quickly enough? How about the battery capacity?
Is this … enough? [SEN] [MASK] … [MASK] [SEN]
q_i
q_{i+1}
The battery … 33%. [SEN] But, … disappointing. [SEN]
a_i
a_{i+1}
Answer Input: The battery capacity has grown by 33%. But, the speed is disappointing.
Bidirectional Contrastive Attention Encoder
Transformer Block
V
K
Q
h
Contrastive A2Q Attention
Add & Norm
Feed Forward
Add & Norm
Transformer Block
Q
K
V
h
Contrastive Q2A Attention
Add & Norm
Feed Forward
Add & Norm
Pretext Tasks
Sentence Retrieval
Is this … enough? [SEN] [MASK] … [MASK] [SEN]
Golden: How about the battery capacity?
Is this … enough? [SEN] [MASK] … [MASK] [SEN]
Sentence Generation
Transferring and Fine-tuning
Step2: Fine-grained Question-Answer Matching
QA Pair Input
Question Input: Is this phone running quickly enough? How about the battery capacity?
Is this … enough? [SEN] How about … capacity? [SEN]
q_i
q_{i+1}
The battery … 33%. [SEN] But, … disappointing. [SEN]
a_i
a_{i+1}
Answer Input: The battery capacity has grown by 33%. But, the speed is disappointing.
QA Pair Encoder
Transformer Block
V
K
Q
h
Multi-head Attention
Add & Norm
Feed Forward
Add & Norm
Transformer Block
Q
K
V
h
Multi-head Attention
Add & Norm
Feed Forward
Add & Norm
Matching Sequence Decoder
Transformer Decoder
q_1 a_1
✓ Matching
…
q_1 a_n
✗ Non- matching
…
q_m a_n
✗ Non-matching

Fig. 2. The framework of our Sentence-aware Contrastive Self-supervised Transfer (SCST) approach to FQAM.

Contrastive Bidirectional Attention Encoder is proposed to unsupervisedly learn the sentence alignment information between QA, which is then transferred to assist the downstream FQAM. This encoder is delicately designed by modifying the basic transformer which is reviewed as follows.

• **Basic Transformer Block** proposed by Vaswani et al. [16] aims to leverage an h-head self-attention to transform each position in the input sequence into a weighted sum of the input sequence itself. Specifically, for each head attention, given an input sequence $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$, this sequence $\mathbf{X}$ is first transformed according to the self-attention as:

$$\text{selfAtt}(\mathbf{X}, \mathbf{X}) = \text{softmax}(\frac{\mathbb{Q}_X^\top \mathbb{K}_X}{\sqrt{d_k}})\mathbb{V}_X \tag{2}$$

where $\mathbb{Q}_X = \mathbf{W}_Q\mathbf{X}$, $\mathbb{K}_X = \mathbf{W}_K\mathbf{X}$ and $\mathbb{V}_X = \mathbf{W}_V\mathbf{X}$ corresponds to queries, keys and values respectively. $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ are trainable parameters. $\sqrt{d_k}$ is the scaling factor. Second, all outputs of the h head attentions are concatenated as the following formula: $\text{multiAtt}(\mathbf{X}, \mathbf{X}) = \mathbf{W}_h[\text{selfAtt}_1(\cdot), ..., \text{selfAtt}_h(\cdot)]^\top$, where $\mathbf{W}_h$ is a trainable parameter. Then a residual connection followed by a normalization operation is used to obtain the further representation as: $\mathbf{H} = \mathbf{LN}(\mathbf{X}+\text{multiAtt}(\mathbf{X}))$. Finally, the output of transformer block is computed as follows:

$$\mathbf{R} = \mathbf{LN}\big(\mathbf{X} + \mathbf{FFN}(\mathbf{H})\big) \tag{3}$$

where $\mathbf{LN}(\cdot)$, $\mathbf{FFN}(\cdot)$ represent layer normalization and feed forward network.

• **Contrastive Bidirectional Attention** is designed to compute contrastive answer-to-question (A2Q) and question-to-answer (Q2A) attentions.

(1) Contrastive A2Q Attention. This attention aims to highlight the words in question aligned with masked sentence $\mathbf{a}_i$ in answer, while filter the contrastive noise in question. Given a QA pair $[\mathbf{Q}, \mathbf{A}]$ which has been processed by Eq. (1), and suppose that $\mathbf{A}$ has been performed sentence masking and masked into the new answer $\hat{\mathbf{A}}$, the contrastive A2Q attention $\text{coAtt}(\hat{\mathbf{A}}, \mathbf{Q})$ is then computed by modifying Eq. (2) in a transformer block as follows:

$$\text{coAtt}(\hat{\mathbf{A}}, \mathbf{Q}) = \big(\mathbb{1} - \text{softmax}(\frac{\mathbb{Q}_{\hat{A}}^{\top}\mathbb{K}_Q}{\sqrt{d_k}})\big)\mathbb{V}_Q \tag{4}$$

where $\mathbb{Q}_{\hat{A}} = \mathbf{W}_{_Q}\hat{\mathbf{A}}$, $\mathbb{K}_Q = \mathbf{W}_{_K}\mathbf{Q}$ and $\mathbb{V}_Q = \mathbf{W}_{_V}\mathbf{Q}$. $\mathbb{1}$ is a unit matrix whose values are all 1. $\mathbb{1} - \text{softmax}(\cdot)$ denotes an operation of calculating opponent attention weights for all words in question. With this operation, we can highlight the part of question dissimilar to unmasked sentences in $\hat{\mathbf{A}}$, since we assume that this part is aligned with the masked sentence $\mathbf{a}_i$. Then, a purified question matrix $\mathbf{R}^{(q)}$ is computed by Eq. (3) for recovering masked answer sentence $\mathbf{a}_i$.

(2) Contrastive Q2A Attention. This attention aims to highlight the words in answer aligned with masked sentence $\mathbf{q}_i$ in question, while filter the corresponding contrastive noise. Given a QA pair $[\mathbf{Q}, \mathbf{A}]$ which has been processed by Eq. (1), and suppose that $\mathbf{Q}$ has been performed sentence masking and masked into the new answer $\hat{\mathbf{Q}}$, contrastive Q2A attention $\text{coAtt}(\hat{\mathbf{Q}}, \mathbf{A})$ is computed by modifying Eq. (2) in another transformer block as:

$$\text{coAtt}(\hat{\mathbf{Q}}, \mathbf{A}) = \big(\mathbb{1} - \text{softmax}(\frac{\mathbb{Q}_{\hat{Q}}^{\top}\mathbb{K}_A}{\sqrt{d_k}})\big)\mathbb{V}_A \tag{5}$$

where $\mathbb{Q}_{\hat{Q}} = \mathbf{W}_{_Q}\hat{\mathbf{Q}}$, $\mathbb{K}_A = \mathbf{W}_{_K}\mathbf{A}$ and $\mathbb{V}_A = \mathbf{W}_{_V}\mathbf{A}$.

Then, a purified answer matrix $\mathbf{R}^{(a)}$ is computed by Eq. (3) for recovering masked question sentence $\mathbf{q}_i$. Note that, since every time only one side in a QA pair is masked, operation $\mathbb{1} - \text{softmax}(\cdot)$ will not be performed in both A2Q and Q2A attention simultaneously.

Pretext Tasks. In our SCST approach, we design sentence retrieval and sentence generation two pretext tasks for pre-training.

• **Sentence Retrieval** aims to select the correct masked sentence from a set of k candidate sentences. Specifically, we first run each sentence independently through a transformer block and obtain the matrix $\mathbf{R}$ of a sentence by Eq. (3). Then, we use a max-pooling to compute the sentence vector as $\boldsymbol{r}_i = \text{pooling}(\mathbf{R})$. Given the sentence set $\mathbf{r} = \{\boldsymbol{r}_, ..., \boldsymbol{r}_l\}$ where l is the number of sentences in all answer texts, this pretext task is to select the masked sentence $\boldsymbol{r}_i$ from this set $\mathbf{r}$. Note that $\mathbf{r}$ is usually very large and a more computationally feasible way is to sample a subset of $\mathbf{r}$ and thus we retrieve negative samples for each masked sentence according to the uniform distribution [12]. Subsequently, we concatenate the question matrix $\mathbf{R}^{(q)}$ and the answer matrix $\mathbf{R}^{(a)}$ to compute the final vector of the masked QA pair as $\boldsymbol{s}_i = \mathbf{W}_s[\text{pooling}(\mathbf{R}^{(q)}); \text{pooling}(\mathbf{R}^{(a)})]$. Here, $\mathbf{W}_s$ is the trainable parameter and $[;]$ denotes the vector concatenation. Finally, the cross-entropy loss of retrieving the masked sentence is given by:

$$\mathcal{L}^{(r)} = -\log p(\boldsymbol{r}_i|\boldsymbol{r}_{1:l}) = -\log(\frac{\exp(\boldsymbol{s}_i{}^\top \boldsymbol{r}_i)}{\sum_{j=1}^{k}\exp(\boldsymbol{s}_j{}^\top \boldsymbol{r}_j)}) \tag{6}$$

- **Sentence Generation** aims to generate the masked sentence token by token. For clarity, we take the generation of masked sentence $\boldsymbol{r}_i$ in answer side as an example. Specifically, we adopt the text generation approach proposed by Mehri et al. [12] to generate the masked sentence $\boldsymbol{r}_i$. Then, let the tokens in $\boldsymbol{r}_i$ be $[w_1, ..., w_N]$, the related likelihood loss of generating $\boldsymbol{r}_i$ is defined as:

$$\mathcal{L}^{(g)} = -\log p(\boldsymbol{r}_i|\boldsymbol{r}_{1:l}) = -\sum_{j}^{N}\log p(w_j|w_{<j}, \boldsymbol{s}_i) \tag{7}$$

where $\boldsymbol{s}_i$ is the final vector of the masked QA pair.

Table 1. Statistics of our constructed datasets. #s/Q (#s/A) and #ch/Q (#ch/A) denote the average number of sentences and Chinese characters in each question (answer) respectively. #m and #n denote the number of *matching* and *non-matching* sentence pairs.

Datasets	#QA	#s/Q	#s/A	#ch/Q	#ch/A	#m	#n
Annotated	32k	3.1	3.9	20.2	28.7	115k	272k
Unannotated	500k	3.4	4.3	21.2	31.2	–	–

2.2 Step2: Fine-Grained QA Matching

In the second step, we formulate FQAM as follows. Given a QA text pair $[\mathbf{Q}, \mathbf{A}]$ where $\mathbf{Q}$ is a sub-question sequence $\{\mathbf{q}_1, ..., \mathbf{q}_m\}$ and $\mathbf{A}$ is a sub-answer sequence $\{\mathbf{a}_1, ..., \mathbf{a}_n\}$, we construct a matching sequence of sub-question and sub-answer pair $[\mathbf{q}_i, \mathbf{a}_j]_{i=1,j=1}^{m,n}$ and obtain mn pairs in total. The goal of FQAM is to predict each pair $\mathcal{P}_t = [\mathbf{q}_i, \mathbf{a}_j]$, $t \in [1, mn]$ inside this matching sequence is *matching* or *non-matching*, which could be seen as a sequence labeling problem. In this way, we design a sequence decoding model to perform FQAM, consisting of a QA pair encoder and a matching sequence decoder as illustrated in Fig. 2.

QA Pair Encoder. To achieve the goal of using the unannotated QA pairs to help FQAM, we transfer and fine-tune the parameters of the contrastive bidirectional encoder pre-trained in the first step to initialize the QA pair encoder in the second step. Suppose that a QA pair $[\mathbf{Q}, \mathbf{A}]$ has been processed by Eq. (1), we first feed this pair to the QA pair encoder. Then, we directly treat the output vector of the closely followed mark "[SEN]" as the vector $\boldsymbol{c}_i^{(q)}$ of sub-question $\mathbf{q}_i$, while the vector $\boldsymbol{c}_j^{(a)}$ is for sub-answer $\mathbf{a}_j$, since transformer possesses the auto-regressive property. Further, we construct the vector-pair sequence $[\boldsymbol{c}_i^{(q)}, \boldsymbol{c}_j^{(a)}]_{i=1,j=1}^{m,n}$ for all sub-question and sub-answer pairs and feed it to the next matching sequence decoder.

Matching Sequence Decoder. Once obtained the vector-pair sequence $[\boldsymbol{c}_i^{(q)}, \mathbf{c}_j^{(a)}]_{i=1,j=1}^{m,n}$, the two vectors inside each vector-pair are concatenated before feeding this sequence into a transformer decoder. Then, a max-pooling is used to transform the transformer output into the final vector $\boldsymbol{v}_t$ for each sub-question and sub-answer pair $\mathcal{P}_t$. Finally, we feed the vector $\boldsymbol{v}_t$ of pair $\mathcal{P}_t$ to a softmax layer for computing the final probabilities of different labels as $p(y|\mathcal{P}_t)$, $y \in \{\textit{matching}, \textit{non-matching}\}$.

2.3 Model Training

In the literature, existing self-supervised learning models [14,17] often adopt the multi-task learning pre-training procedure. In the first step, our contrastive self-supervised model also uses the multi-task learning framework to jointly learn the sentence retrieval and generation tasks.

Then, the joint loss function for our self-supervised model in the first step is defined as: $\mathcal{L} = \mathcal{L}^{(r)} + \mathcal{L}^{(g)}$, where $\mathcal{L}^{(r)}$ and $\mathcal{L}^{(g)}$ are the losses for sentence retrieval and sentence generation respectively. In the second step, with the label probabilities of each sub-question and sub-answer pair $\mathcal{P}_t$, we minimize the negative log-likelihood loss of a QA text pair for the FQAM task as: $\mathcal{L}^{(f)} = -\sum_{t=1}^{T} \log p(y_t|\mathcal{P}_t)$, where $T = mn$ is the number of pair $\mathcal{P}_t$ in a QA text pair. y_t is the ground-truth label for $\mathcal{P}_t$.

3 Experimental Settings

Data Settings. In this study, we collect 532k Chinese QA text pairs from the e-commerce platform *Taobao*[3] on which the many-to-many style QA pairs are rather pervasive. Then, we randomly select 32k QA text pairs to perform manual annotation for FQAM, and treat the rest 500k QA pairs as the large-scale unannotated dataset for performing pre-training. Note that, we adopt the well-studied sentence segmentation tool[2] to segment each question and answer text into sentences and then directly treat each sentence as the sub-question or sub-answer to perform annotation for FQAM. We believe the above process can make FQAM more practical to large-scale application and can also make our SCST approach more flexible and scalable, since most of sub-questions and sub-answers are located in one sentence and more importantly deciding the boundary of each sub-question and sub-answer is rather difficult and not practical for real-world applications. The statistics of our constructed datasets are shown in Table 1. Then, we randomly split the ***Annotated*** dataset into train, dev, and test sets with the ratio of 7:1:2 for FQAM.

[3] http://www.taobao.com.

Table 2. Performance comparison of various approaches to FQAM, where P, R, F1 and Acc. represent Precision, Recall, Macro-F1 and Accuracy, respectively. Besides, "+ QA Pairs" denotes BERT and RoBERTa are further trained with our 500k unannotated QA pairs, and then fine-tuned by casting these unannotated QA pairs as a one-to-one QA matching task.

	Approaches	P	R	F1	Acc.
Single	BiMPM [7]	68.3	69.2	68.7	69.4
	ESIM [18]	68.7	68.9	68.8	71.2
	RE2 [19]	68.9	72.3	70.6	71.7
	BERT [5]	73.1	74.7	73.9	75.1
	BERT + QA Pairs	75.2	76.8	76.0	77.5
	RoBERTa [15]	73.9	75.9	74.9	76.3
	RoBERTa + QA Pairs	76.3	77.5	76.9	77.1
Sequence	BiMPM [7]	69.7	70.2	69.9	71.1
	ESIM [18]	71.1	71.7	71.4	71.5
	RE2 [19]	71.7	73.7	72.7	73.8
	BERT [5]	74.5	76.2	75.3	76.5
	BERT + QA Pairs	75.3	77.8	76.5	77.2
	RoBERTa [15]	74.8	77.4	76.1	78.1
	RoBERTa + QA Pairs	75.2	**79.5**	77.3	78.3
Ours	SCST (only using Step2)	72.3	74.2	73.2	74.1
	SCST	**78.3**	79.2	**78.7**	**80.5**

Baselines. For FQAM, we implement some approaches [5,7,15,18,19] as baselines. Concretely, the baselines contain **Single** and **Sequence** two groups. **Single** group is to treat the matched sub-question and sub-answer pair as positive samples while the mismatched pairs as negative samples and then directly use the baseline approach to perform binary classification for each sample. **Sequence** group is to treat FQAM as a sequence labeling problem. Specifically, we first use a shared encoder to encode each sub-question and sub-answer pair, and then use an extra transformer decoder to decode matching sequence like our SCST approach. Here, different approaches shown in Table 2 are adopted as the shared encoder for comparison.

Implementation Details. All experiments adopt BERT-Base (Chinese) and RoBERTa-Base (Chinese). In the first step of SCST, contrastive bidirectional attention encoder is first initialized with the last layer of the released RoBERTa, and then pre-trained with 500k unannotated pairs. Besides, in each experiment, we measure the runtime 10 times and average the results.

Evaluation Metrics. The performance is evaluated with *Precision* (P), *Recall* (R), *Macro-F1* (F1) and *Accuracy* (Acc.). Besides, the paired t-test[4] is used to evaluate the significance of the performance difference of two approaches.

4 Results and Discussion

Experimental Results. Table 2 shows the performances of different approaches to FQAM. From this table, we can see that, two large-scale pre-trained self-supervised models, i.e., **BERT** and **RoBERTa**, significantly outperform (p-value < 0.05) the traditional attention based text matching approaches, i.e., **BiMPM**, **ESIM** and **RE2**. This confirms the powerful transfer ability of pre-trained models for downstream tasks including FQAM. When further trained and fine-tuned with unannotated QA pairs, **BERT** and **RoBERTa** can consistently achieve better performance. This encourages us to leverage the large-scale unannotated QA pairs to boost the performance of FQAM whose annotation is rather time-consuming and labor-intensive.

Furthermore, when treating FQAM as a sequence labeling problem, all approaches in the second group **Sequence** perform better than their corresponding approaches in the first group **Single**. This is reasonable. Take Fig. 1 as an example, if Q2 has been predicted to be matched with A1, the pair Q2 and A3 is more possibly to be predicted as *non-matching*. This dependency information could be well captured by the transformer decoder. These results encourage us to treat FQAM as a sequence labeling problem instead of binary classification.

In contrast, our **SCST** approach outperforms all above baselines and even significantly outperforms (p-value < 0.05) the strong pre-trained model **RoBERTa + QA Pairs**. This justifies the effectiveness of SCST in unsupervisedly capturing the sentence alignment information between QA. Impressively, compared with **SCST** which removes Step1, **SCST** achieves the improvement of 5.5% in terms of *Macro-F1* and 6.4% in terms of *Accuracy*. Significance test shows that these improvements are significant (p-value < 0.05). This again justifies the importance of leveraging the unannotated QA pairs to help FQAM and the effectiveness of SCST in performing pre-training with these QA pairs.

Ablation Study. Table 3 shows the ablation results to evaluate the contribution of each key component for SCST. From Table 3, we can see that: **1)** Fine-tuning bidirectional contrastive attention encoder with the annotated FQAM dataset can improve the Acc. by 1.4%. **2)** Incorporating contrastive Q2A and A2Q attention into SCST can improve the Acc. by 2.3% and 1.2%. This indicates that treating QA text pairs as parallel units and considering the bidirectional matching information is helpful. **3)** Using the pretext task sentence retrieval and sentence generation can improve the Acc. by 1.8% and 1.2%. **4)** Using RoBERTa to initialize the transformer blocks inside contrastive bidirectional encoder can improve the Acc. by 3.3%. This is reasonable since the data for pre-training RoBERTa from scratch is much larger than our 500k QA pairs.

[4] https://www.scipy.org/.

Table 3. Ablation study for our SCST approach.

Approaches	P	R	F1	Acc.
SCST	**78.3**	**79.2**	**78.7**	**80.5**
– Fine-tuning with FQAM	77.5	78.2	77.8	79.1
– Contrastive Q2A Attention	75.3	76.9	76.1	78.2
– Contrastive A2Q Attention	76.5	78.8	77.6	79.3
– Sentence Retrieval	76.9	78.1	77.5	78.7
– Sentence Generation	77.2	78.7	77.9	79.3
– Initialized with RoBERTa	74.3	76.9	75.6	77.2

Table 4. Performance comparison of SCST and four strong baselines on the purified test set where all the QA pairs have the reversed-order phenomenon.

	Approaches	P	R	F1	Acc.
Single	BERT + QA Pairs	63.4	66.2	64.8	65.7
	RoBERTa + QA Pairs	64.3	66.3	65.3	66.8
Sequence	BERT + QA Pairs	67.5	69.1	68.3	70.2
	RoBERTa + QA Pairs	68.2	70.3	69.2	70.6
Ours	**SCST**	**71.9**	**74.3**	**73.1**	**74.2**

Robustness Analysis of Transfer. To investigate the transfer ability of our SCST approach, we evaluate it and three strong baselines on different numbers of training examples for FQAM. As shown in Fig. 3, **1)** SCST can significantly improve the performance from 34.7% to 46.2% on small size of training data (only 10% training data) compared to RE2 without pre-training. **2)** SCST consistently performs better than all pre-training based approaches, i.e., BERT and RoBERTa, on various sizes of training data. These justify the robustness of SCST in terms of transfer to FQAM.

Effectiveness Analysis of Sequence Decoding with Reversed-order Samples. To evaluate the effectiveness of our proposed matching sequence decoder for FQAM and enhance the task difficulty, we construct a cleaned and more difficult test set, of which all the QA pairs have the reversed-order phenomenon (e.g. Q1 $\xrightarrow{\checkmark}$ A2 while Q2 $\xrightarrow{\checkmark}$ A1 in Fig. 1). Table 4 shows the results on the cleaned test set. From which, we can see that the approaches in the **Sequence** group perform much better than **Single**. This justifies the effectiveness of our proposed sequence decoder and encourages us to consider FQAM as a sequence labeling problem. Moreover, our SCST approach still outperforms (p-value < 0.05) all the four strong baselines, again justifying the robustness of SCST.

Error Analysis of SCST. We randomly analyze 200 error cases of SCST and categorize them into 4 types. **1)** 48% of errors are due to the ambiguous

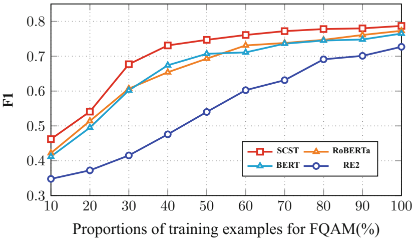


Fig. 3. Comparison of SCST and three baselines from the **sequence** group, trained with different proportions of training examples for FQAM, where BERT and RoBERTa have been further trained and fine-tuned with 500k QA pairs.

reference and complicated semantics. An example is "Q: *How is screen? How about battery? Is the speed fast?* A: *Not good. Anyway, screen is nice. The speed is impressive.*". SCST fails to capture the related sub-question of the sub-answer "*not good*" with the ellipsis of reference; **2)** 26% of errors are due to the lack of external knowledge. An example is "Q: *Can it run Tensorflow?* A: *Don't worry, it has perfect hardware.*". SCST incorrectly predicts *non-matching.* **3)** 14% are due to the long length (e.g., more than 10 sentences) of questions or answers. **4)** 12% are due to fuzzy boundaries and incorrect segmentation of sub-questions or sub-answers. An example is "Q: *How about price, quality and service?* A: *Not expensive. Good quality. But the after-sale service is terrible.*". SCST incorrectly predicts *non-matching* for sub-answer "*not expensive*", inspiring us to perform question/answer decomposition [20] for FQAM.

5 Related Work

Text Matching aims to predict whether a given text pair has similar semantics. Dominant paradigms for text matching focus on leveraging the multi-perspective matching model [7] and attention based neural networks [8] to capture the matching relationship. QA matching is a sub-task of text matching, which aims to predict the matching relationship between a QA pair. Recently, Shen et al. [9] propose to compute the co-occurrence probability in QA pairs. Wang et al. [10] also propose a fine-grained QA matching task, but their task is limited to the one-to-many scenario. Unlikely, we extend text matching to a more general many-to-many matching paradigm and propose a new FQAM task. To our best knowledge, this is the first attempt to address this task.

Self-supervised Learning uses pretext tasks to replace the manually annotated labels with "pseudo-label" obtained from the raw data. Actually, BERT [5] is a typical self-supervised model, adopting the word masking to perform pretraining. Besides, Wu et al. [11] and Mehri et al. [12] leverage utterance masking to detect the utterance order inside dialogue for pre-training. Recently, contrastive learning [13] has attracted much attention, which incorporates negative pairs selection and contrastive losses to perform self-supervised learning.

Unlike all the above studies, we propose a new contrastive self-supervised learning framework, not relying on either negative pairs or contrastive losses, to unsupervisedly capture the sentence alignment information between QA.

6 Conclusion

In this paper, we propose a new FQAM task and build a high-quality annotated corpus for this task. On this basis, we propose a sentence-aware contrastive self-supervised transfer (SCST) approach. The basic idea of SCST is to leverage large-scale unannotated QA pairs to help FQAM with limited labeled data. Empirical studies show that SCST significantly outperforms two SOTA pre-trained baselines in FQAM. In our future work, we would like to solve other challenges in FQAM, such as the ellipsis of reference and the lack of external knowledge. Furthermore, we would like to investigate our SCST approach in other tasks whose inputs are also parallel units, e.g., cross-lingual analysis.

Acknowledgements. This work was supported by three NSFC grants, i.e., No. 62006166, No. 62076176 and No. 61976146. This work was also supported by a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). Also, we would like to thank the anonymous reviewers for their helpful comments.

References

1. Trischler, A., Ye, Z., Yuan, X., He, J., Bachman, P.: A parallel-hierarchical model for machine comprehension on sparse data. In: Proceedings of ACL 2016, Berlin, Germany (2016)
2. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: Proceedings of EMNLP 2018, Brussels, Belgium, pp. 2369–2380 (2018)
3. Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., Zhou, M.: SuperAgent: a customer service chatbot for E-commerce websites. In: Proceedings of ACL 2017, Vancouver, Canada, pp. 97–102 (2017)
4. Wang, J.C., et al.: Sentiment classification in customer service dialogue with topic-aware multi-task learning. In: Proceedings of AAAI 2020, New York, USA, pp. 9177–9184 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL 2019, Minneapolis, pp. 4171–4186 (2019)
6. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguist. **8**, 64–77 (2020)
7. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: Proceedings of IJCAI 2017, Melbourne, Australia, pp. 4144–4150 (2017)
8. Rao, J., Liu, L., Tay, Y., Yang, H.W., Shi, P., Lin, J.: Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In: Proceedings of EMNLP 2019, China, pp. 5369–5380 (2019)

9. Shen, Y., Rong, W., Jiang, N., Peng, B., Tang, J., Xiong, Z.: Word embedding based correlation model for question/answer matching. In: Proceedings of AAAI 2017, San Francisco, California, USA, pp. 3511–3517 (2017)
10. Wang, L., et al.: One vs. many QA matching with both word-level and sentence-level attention network. In: Proceedings of COLING 2018, New Mexico, USA, pp. 2540–2550 (2018)
11. Wu, J., Wang, X., Wang, W.Y.: Self-supervised dialogue learning. In: Proceedings of ACL 2019, Italy, pp. 3857–3867 (2019)
12. Mehri, S., Razumovskaia, E., Zhao, T., Eskénazi, M.: Pretraining methods for dialog context representation learning. In: Proceedings of ACL 2019, Florence, Italy, pp. 3836–3845 (2019)
13. Yang, Z., Cheng, Y., Liu, Y., Sun, M.: Reducing word omission errors in neural machine translation: a contrastive learning approach. In: Proceedings of ACL 2019, Florence, Italy, pp. 6191–6196 (2019)
14. Wang, S., Che, W., Liu, Q., Qin, P., Liu, T., Wang, W.Y.: Multi-task self-supervised learning for disfluency detection. In: Proceedings of AAAI 2020, New York, NY, USA, pp. 9193–9200 (2020)
15. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. CoRR (2019)
16. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS 2017, Long Beach, CA, pp. 5998–6008 (2017)
17. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: Proceedings of ICCV 2017, Venice, Italy, pp. 2070–2079 (2017)
18. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: Proceedings of ACL 2017, Vancouver, Canada, pp. 1657–1668 (2017)
19. Yang, R., Zhang, J., Gao, X., Ji, F., Chen, H.: Simple and effective text matching with richer alignment features. In: Proceedings of ACL 2019, Florence, Italy, pp. 4699–4709 (2019)
20. Perez, E., Lewis, P.S.H., Yih, W., Cho, K., Kiela, D.: Unsupervised question decomposition for question answering. CoRR (2020)

KARN: Knowledge Augmented Reasoning Network for Question Answering

Lishuang Li(✉), Huxiong Chen, Xueyang Qin, Zehao Wang, and Jiangyuan Dong

School of Computer Science and Technology,
Dalian University of Technology, Dalian 116023, Liaoning, China
lils@dlut.edu.cn

Abstract. Question answering with information from pre-trained language models (LMs) and knowledge graphs (KGs) presents the following problems. First of all, some information in the knowledge graph is irrelevant to the question and answer. Then, the influence of edge information on node representation is ignored. To address these issues, we propose a Knowledge Augmented Reasoning Network (KARN), which utilizes external knowledge and contextual information to remove redundant entities, while enhancing the feature representation of nodes by optimizing the weight of edges. Specifically, we first use context and external knowledge to score the relevance of nodes and tasks, and then for nodes with low scores, we delete nodes and their related edges. Furthermore, we consider the node's global information and external knowledge in the edge weight calculation, making the calculation of edge weight scores more comprehensive. We conduct experiments on three datasets in the fields of commonsense reasoning (*i.e.*, CommonsenseQA, OpenbookQA) and medical question answering (*i.e.*, MedQA-USMLE), and the experimental results verify the effectiveness of our method.

Keywords: Question answering · Knowledge augmented · Reasoning network

1 Introduction

Question answering (QA) based on the knowledge graph means responding to natural language questions using the knowledge contained in the knowledge graph. The "simple question" in question answering, which only includes a head entity and a relationship, can be perfectly solved using the knowledge graph. But when the problem becomes complex, there are multiple entities, multiple relations, multi-hop reasoning, constraint relations, numerical operations, or some combination of the above. Relying solely on knowledge graphs can no longer meet the needs of reasoning. For complex question answering, it is usually necessary to introduce external knowledge and context information, obtain a subgraph of the

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 629–641, 2023.
https://doi.org/10.1007/978-3-031-44693-1_49

knowledge graph, and perform interpretable reasoning on the paths of subgraphs through graph neural networks (GNN).

In general, question answering based on reasoning can be divided into two groups. One uses pre-trained language models to encode knowledge implicitly but fails to provide interpretable predictions. The other combines the language model (LM) and knowledge graph (KG) to explain the rationality of the final answer by inferring paths in the knowledge graph. For example, MHGRN [3] uses the method of LM+KG to perform multi-hop, multi-relational reasoning on subgraphs extracted from knowledge graphs. The proposed reasoning module unifies path-based reasoning methods and the graph neural network for better interpretability and scalability. KagNet [9] grounds a question-answer pair from the semantic space to the knowledge-based symbolic space as a schema graph, a related subgraph of external knowledge graphs, and scores answers with graph representations. Similarly, some approaches, such as [12,19], apply LM to context and graph neural networks to KG, obtaining limited inference. Based on this, Yasunaga et al [22] exploit LM to estimate the importance of KG nodes and QA context, and update the representations of nodes through graph neural networks to enhance nodes' features effectively. Furthermore, GREASELM [23] fuses encoded representations from pre-trained LMs and graph neural networks(GNN) on multi-layer modality interaction operations, enabling information from two modalities to propagate to the other. However, these methods ignore two critical issues: the existence of redundant entities in the knowledge graph and the insufficient attention to the influence of edges on node features in the path of the knowledge graph.

To address the above issues, we propose a new model (KARN) that can combine LM and KG for interpretable reasoning. For the redundant entities in the subgraph, we propose to combine the contextual knowledge and external knowledge to score the relevance of the nodes in the subgraph, and delete the less relevant nodes and related edges according to this score. Considering the influence of edge weights when updating the node features in the subgraph, we utilize global information and external knowledge as part of the factors affecting edge weights for edge weight calculation, further enhancing node feature representation. After improving the node features, the model obtains more information from the nodes, and the reasoning of the path is more accurate, that is, both types of knowledge help augment reasoning. The experiment proves that KARN outperforms the previous LM+KG model of the same size on the three datasets of CommonsenseQA, OpenbookQA, and MedQA-USMLE. The contributions of our work are summarized as follows:

- We propose a knowledge augmented graph reasoning network (KARN) based on LM+KG, and the experimental results on three datasets, CommonsenseQA, OpenbookQA, and MedQA-USMLE, demonstrate the effectiveness of our method.
- We propose a novel knowledge graph pruning strategy, comprehensively considering the question-answering context and external knowledge, which makes the pruning process more reasonable.

- To the best of our knowledge, we increase the global information and some concept features to calculate edge weights in the GNN reasoning module for the first time to optimize the weight of edges and fully display the role of edges in the graph reasoning module.

2 Problem Statement

Our goal is to answer natural language questions using pre-trained language model LM and knowledge graph KG as well as external knowledge (Wiktionary). For a given natural language problem, we use the function $f_{enc}(x)$ to map the input text information x to the vector representation of the text through the language model LM, and use $f_{comp}(x)$ to calculate the relevance score of x. Specifically: we concatenate all the information of each entity with all the information of the context, and then use the pre-trained model to calculate their similarity. The knowledge graph KG is defined as $G = (V, E)$, where V is a collection of entities, $E \subseteq V \times R \times V$ is a collection of edges connecting nodes in V, and R represents a collection of relation types.

Given a question q and the answer a, we link the question entity V_q and the answer entity V_a to the given knowledge graph $(V_q, V_a \subseteq V)$, and use the context and node concept knowledge to get the subgraph. Then the subgraph is pruned. Finally, we use GNN to infer the pruned subgraph to get the answer.

3 Method

As shown in Fig. 1, given a question q and answer a. We combine the question and answer as contextual knowledge (CK). KARN works as follows. First, we use the pre-trained LM to obtain the context representation (CK^{LM}). Then, we retrieve the knowledge graph according to the external knowledge and context to get the original QA subgraph (G_i). Second, we calculate the correlation score between each node and the task in the original subgraph according to the context and external knowledge. We then delete redundant nodes in G_i to obtain the subgraph (G_{sub}). Then, to enhance node features, we improve the way of updating node features in GNN by optimizing edge weights. Finally, we use the pre-trained QA context features, pooled inference paths, and the QA context features aggregated by the GNN model to make final predictions.

3.1 Graph Construction

To facilitate path reasoning, we retrieve QA subgraphs from knowledge graphs based on contextual knowledge. Firstly, link the question and answer entities in the question answer context to the knowledge graph KG, and start from the question entity to retrieve all entities on the 3-hop path that terminate with any answer option. We get a subgraph with many entities and term it $G_i = (V, E)$, where $V = \{V_q, V_a, V_c\}$ and E is the set of edges. V_q, V_c, and V_a are the

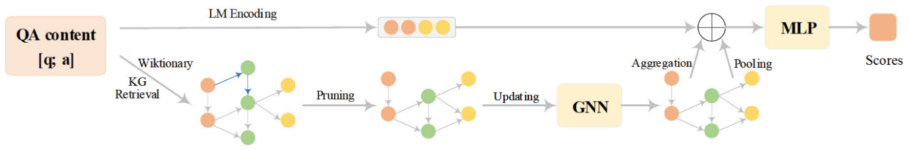


Fig. 1. The overall framework of KARN. According to the question and answer context, we retrieve the knowledge graph, obtain external knowledge, calculate the node correlation score, and then prune the subgraph (after pruning, a green node and its two blue edges are deleted from the graph). The inference is performed on the pruned subgraph with GNN, and node features are updated according to edge and node weights. Finally, use the three parts of context representation, GNN context, and pooled reasoning path to find the path with the highest score. (Color figure online)

set of question nodes, intermediate path nodes, and answer nodes, respectively (corresponding to the node colors in Fig. 1, brown, green, and yellow). Then, we use Wiktionary to extend the concept of each node in the subgraph. We select the most suitable concept explanation of nodes in the following order from Wiktionary: "*original form > lemma form by Spacy > base word*". Finally, each node v_i has a concept description v_i^c, where $v_i \in V$.

3.2 Pruning Knowledge Graph

Many nodes on the KG subgraph G_i may be irrelevant under the current QA context. These nodes may lead to over fitting or introduce many unnecessary reasoning paths. For example, the word "people" generally has little effect on reasoning. We retrieved 1,500 QA data in the three datasets respectively. When we consider 3-hop neighbors, we retrieve a subgraph with $|V_{sub}| \approx 480$ nodes on average, where V_{sub} is the total number of nodes in the subgraph. This needs to use context and external knowledge to score the relevance of nodes to the question answering task, and remove low-scoring nodes.

To make the relevance score more reasonable, we propose concatenating the context CK and the concept description of the context node $\{Q_c, A_c\}$ as the task information, and the nodes and their descriptions as the node information. Then, we use the pre-trained language model to score the relevance of each node information and task information. The formula is as follows:

$$\begin{aligned} v_f &= [(CK; \{Q_c, A_c\}), (v_i^c, v_i)], \\ \rho_{v_f} &= f_{comp}(f_{enc}(v_f)), \end{aligned} \tag{1}$$

where ρ_{v_f} denotes the relevance score. v_f means to concatenate the task information with node information, also called global information.

We keep the top 200 nodes and then prune the remaining nodes. If the ρ_{v_f} of the node is below 200, we delete the node and its related edges. As shown in Fig. 1, the blue edges and a green node are removed. This practice reduces the reasoning path. In this way, we finally get a multi-relational knowledge graph subgraph (G_{sub}) with a relatively high correlation coefficient with the context, where $G_{sub} = (V_g, E_g)$, where $V_g = (V_q, V_a, V_g)$ and E_g is the set of the remaining edges. V_g is the set of remaining intermediate path nodes.

3.3 GNN Architecture

To reason about the G_{sub} obtained in Sect. 3.2 and to make full use of context information and external knowledge information in graph reasoning, we propose a GNN model based on RGAT [4]. On the basis of RGAT, we increase the influence of edges on node weights, and introduce context information and external knowledge into edge weight calculations. Node features are finally jointly updated by neighbor nodes and edges (As shown in Fig. 2).

Enhanced Node Feature. In this subsection, we adopt the features of neighbor nodes to enhance the feature representation of central node $v_i \in V_g$. Specifically, we first calculate the attention scores N_{ij}^{lk} of node v_i and neighbor node v_j. Then according to this score, the features of neighbor nodes are aggregated, and the initially enhanced node features $h_{N_{it}}^{l+1}$ are obtained. The formula for the whole process is as follows:

$$\begin{aligned}
\overrightarrow{v_i} &= f_{enc}(v_i),\\
N_{ij}^{lk} &= \phi(\overrightarrow{v_i}, \overrightarrow{v_j}),\\
h_{N_{it}}^{l+1} &= ||_{k=1}^{K} \textstyle\sum_{j\in N_i} N_{ij}^{lk} W_k^l h_j^l,
\end{aligned} \tag{2}$$

where $\overrightarrow{v_i}$ is the vectorized representation of v_i. ϕ means the following operations: first, define $\overrightarrow{a}$, whose dimension is twice the feature dimension of the mapped node v_i, concatenate the two column vectors $(\overrightarrow{v_i}, \overrightarrow{v_j})$ by column, and then perform dot multiplication with $\overrightarrow{a}$ to calculate the inner product. $||_{k=1}^{K} x_i$ represents the concatenation of vectors from x_1 to x_k. W_k^l denotes the transformation matrix parameter of the corresponding neighbor node feature, and h_j^l is the corresponding neighbor node feature. N_i is the neighbor node of v_i. K represents the number of attention heads, meaning that K independent attention mechanisms are used to calculate the hidden state and connect their features.

Calculation of Relation Weight. G_{sub} is a multi-relational graph. For nodes, different relations should have different effects. In this subsection, first, we adopt the combination of external knowledge and contextual nodes to obtain the attention score of context to relation R_{rv}. Then get the attention scores of the relation and its dependent nodes. Finally, we weight the two parts to obtain the final relation weight β_R. Specifically, the calculation formula is as follows:

$$\begin{aligned}
&\beta_{hv} = \sum_{j=1}^{n} \phi(\overrightarrow{v_h}, \overrightarrow{v_i}),\\
&\beta_{tv} = \sum_{j=1}^{n} \phi(\overrightarrow{v_t}, \overrightarrow{v_i}),\\
&\beta_{qv} = \phi(\overrightarrow{v_h}, \overrightarrow{Q_c||A_c})\rho_{v_h} + \phi(\overrightarrow{v_t}, \overrightarrow{Q_c||A_c})\rho_{v_t}, R_{rv} = \tfrac{\beta_{hv}+\beta_{tv}+\beta_{qv}}{2(n+1)},\\
&R^{lm} = sigma(\theta(R_{ij}W_{m1} + b_{m1})W_{m2} + b_{m2}),\\
&\beta_R = \alpha_1 \frac{\exp(R^{lm})}{\sum_{j=1}^{N_i} \exp(R^{lm})} + \alpha_2 R_{rv},
\end{aligned} \tag{3}$$

where n represents the number of context nodes, sigma is the activation function sigmoid, ρ_{v_h} and ρ_{v_t} means the relevant scores of the head and tail nodes,

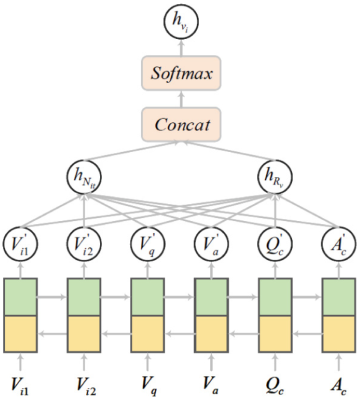


Fig. 2. Enhancement method of node features.

respectively. θ is the activation function *relu*. v_h and v_t mean the head and tail nodes of the relation, respectively. β_{hv} is the total attention score of the head node and the context node. β_{tv} is the total attention score of the tail node and the context node. R_{ij} represents the relational embedding between node v_i and node v_j. α denotes a feature parameter, which adjusts the score of two-part relation attention, $\alpha \in (0, 1)$, and $\alpha_1 + \alpha_2 = 1$.

Node Features. After the weight of the relation is calculated, the final node feature $h_{v_i}^{l+1}$ can be calculated. First, we obtain the node features $h_{R_v}^{l+1}$ according to the relation weights. In order to achieve the effect of enhancing node features, we concatenate $h_{N_{it}}^{l+1}$ and $h_{R_v}^{l+1}$, and the final node features can be obtained through a full connection and activation function. The mathematical representation is as follows:

$$\begin{aligned} h_{R_v}^{l+1} &= ||_{m=1}^{M} \textstyle\sum_{j \in N_i} \beta_R W_l^m h_j^l, \\ R_v^{l+1} &= h_{N_{it}}^{l+1} + h_{R_v}^{l+1}, \\ h_{v_i}^{l+1} &= \theta(W_{l+1} R_v^{l+1} + b_{l+1}), \end{aligned} \tag{4}$$

where m represents the number of the relational attention heads. W_m^l is the transformation matrix parameter corresponding to the neighbor node feature.

3.4 Learning and Inference

In previous work, some methods only link contextual information into question entity and answer entity [22]. However, our method introduces the concept information of the node, which makes the relation between the node and the context more intimate, so that the context can act on any node, thereby enhancing the feature representation of the node. Using the node features enhanced by context and external knowledge for path reasoning will make the reasoning more reasonable.

Given a question answering context CK, the probability that a is the correct answer can be calculated as $P(a|Q) \propto \exp(MLP(CK^{LM}, CK^{GNN}, \Delta))$, where

Δ represents the pooling of node features in the inference path. CK^{GNN} means the feature of the context updated by GNN obtained by aggregating the context features of all nodes on the path. The context information at this time has a global effect. CK^{LM} denotes the features of the context passing through the language model. The specific formula is as follows:

$$\begin{aligned} CK^{LM} &= f_{enc}(CK), \\ CK^{GNN} &= \tfrac{1}{r}\textstyle\sum_{i=1}^{r} V_i^f, \end{aligned} \tag{5}$$

where r means the number of nodes on the path, V_i^f represent the context feature in each node.

4 Experiment

4.1 Datasets and External Knowledge

Datasets. We evaluate KARN models on three datasets, CommonsenseQA (CSQA) [18], OpenBookQA (OBQA) [13], MedQA-USMLE [5]. CommonsenseQA is a dataset of 5-way multiple-choice responses to 12,102 questions. OpenBookQA is a 4-choice multiple-choice QA task that requires reasoning using fundamental scientific knowledge and contains 5,957 questions. We split the official 2018 data [14]. MedQA-USMLE is a 4-way multiple-choice QA task that requires biomedical and clinical knowledge. The dataset contains 12,723 questions. We split the raw data using the method of [22].

External Knowledge. We use a general domain knowledge graph ConceptNet [17] as an external knowledge base for our common sense question answering datasets Commonsense-QA and OpenbookQA. For MedQA-USMLE, we use a knowledge graph built by Yasunaga [22] in 2021. The knowledge graph contains 9958 nodes and 44561 edges. Wiktionary includes different vocabularies in more than 170 languages, which can fully satisfy our three datasets' conceptual interpretation of question-answering entities.

4.2 Baselines

Fine-Tuned LMs & LM+KG Models. To study the effect of using knowledge graphs and external dictionary bases as external knowledge sources, we compare our method with knowledge-agnostic vanilla fine-tuned LMs. For the CommonsenseQA, OpenBookQA, and MedQA-USMLE datasets, we use language models RoBERTa [11], AristoRoBERTa [2], and SapBERT [10] for processing, respectively. We also evaluate KARN's ability to leverage external knowledge (knowledge graph and external dictionary base) and context for question answering by comparing with existing LM+KG methods. The existing LM+KG methods are:KagNet [9], MHGRN [3] QA-GNN [22], GREASELM [23], etc. Under the LM+KG method, the performance of GREASELM is the best above. The biggest difference between our method and these baseline models is that we introduce edge weights and incorporate external knowledge information when computing node features to enhance node feature representation.

Table 1. Performance comparison on CommonsenseQA in-house split. As the official test is hidden, here we report the in-house Dev (IHdev) and Test (IHtest) accuracy. Experiments are controlled using same seed LM.

Model	IHdev (%)	IHtest (%)
RoBERTa-large	73.1 (±0.5)	68.7 (±0.6)
+ RGCN [1]	72.7 (±0.2)	68.4 (±0.7)
+ GconAttn [19]	72.6 (±0.4)	68.6 (±1.0)
+ KagNet [9]	73.5 (±0.2)	69.0 (±0.8)
+ RN [16]	74.6 (±0.9)	69.1 (±0.2)
+ MHGRN [3]	74.5 (±0.1)	71.1 (±0.8)
+ QA-GNN [22]	76.5 (±0.2)	73.4 (±0.9)
+ GREASELM [23]	78.5 (±0.5)	74.2 (±0.4)
+ **KARN (our)**	**79.4 (±0.9)**	**76.4 (±0.3)**

Table 2. Test accuracy comparison to public OpenBookQA model implementations. UnifiedQA (11 B params) are 30x large than ours.

Model	Test (%)	#Params
ALBERT+KB [9]	81.0	∼235M
HGN [21]	81.4	≥361M
DEKCOR [20]	82.4	∼
QA-GNN [22]	82.8	∼360M
T5 [15]	83.2	∼3B
UnifiedQA [7]	**87.2**	**∼11B**
GREASELM [23]	84.8	∼359M
KARN (our)	85.2	∼360M

4.3 Implementation and Training Details

We set the dimension of node embeddings in the GNN module to 200 and the number of layers to 5, with a dropout rate of 0.2 applied to each layer. We train the model with one GPU (GeForce RTX 3090), which takes about 8 h. We set the batch size from {64, 128}, learning rate for the LM module from {1e−5, 2e−5, 3e−5, 5e−5}, and learning rate for the GNN module from {5e−3, 1e−3, 2e−3}.

Table 3. Test Accuracy comparison on MedQA-USMLE.

Model	Test (%)
BERT-base [6]	34.3
BioBERT-base [8]	34.1
BioBERT-LARGE [8]	36.7
SapBERT-Base [10]	37.2
QA-GNN [22]	38.0
GREASELM [23]	38.5
KARN (ours)	**43.9**

Table 4. Test Accuracy comparison on OpenBookQA (The same seed LM).

Model	Test (%)
AristoRoBERTa	78.4
+RGCN [1]	74.6
+RN [16]	75.6
+MHGRN [3]	80.6
+QA-GNN [22]	82.8
+GREASELM [23]	84.8
KARN (ours)	**85.2**

Table 5. Performance of KARN on the CommonsenseQA IH-dev set on negative questions.

Model	Negation (%)
RoBERTa-large (w/o KG)	63.8
QA-GNN [22]	66.2
GREASELM [23]	69.9
KARN (our)	**70.3**

4.4 Main Results

Table 1 shows the quantitative results of our KARN approach on CommonsenseQA. The performance of KARN is 7.7% higher than fine-tuned LMs and 2.2% higher than the existing best LM+KG model GREASELM.

The experimental results on the OpenBookQA are shown in Table 2. We can see that our KARN outperforms the previous best LM+KG model (GREASELM) by 0.4%. The parameter of the best model (UnifiedQA) is thirty times larger than ours. KARN performance is the best if the parameters are similar.

Table 3 shows the accuracy of KARN method on MedQA-USMLE, the performance of our model on the test set is 6.7% higher than LMs also using SapBERT-Base, 7.2% higher than the best performing BioBERT-LARGE in BioBERT, and better than the prior best LM+KG model (GREASELM) is 5.4% higher. The possible reason is that external knowledge descriptions of entities in the medical dataset often correlate with answers, so the improvement is noticeable.

Table 4 represents that when we use the same pre-trained model, KARN still performs well on OpenbookQA compared to other models. Experimental results demonstrate the robustness of our KARN.

Table 5 shows the performance of our KARN approach when dealing with negative problems on CommonsenseQA. Clearly, KARN outperforms the same architecture LM+KG models (QA-GNN and GREASELM) in coping with negative issues, demonstrating its excellent structured reasoning ability.

4.5 Ablation Experiment and Analysis

We conduct ablation experiments on our model on the dev set of the CSQA and ablate the three parts of whether to introduce the external knowledge (Wiktionary), the calculation of edge weights in GNN, and the Clipping Knowledge Graph to verify our work effectiveness further.

External Knowledge (Wiktionary). The first ablation experiment we performed is for using external knowledge or not. It can be seen from Table 6 that when we do not use contextual knowledge at all, the performance of the model drops directly by 2.2%. If we only use external knowledge without integrating with the context, the performance of the model is not as good as that without

Table 6. Ablation experiments.

Ablation Part	Dev-acc (%)
w/o external knowledge	77.2
External knowledge (w/o fusing)	76.8
w/o edge weights	76.5
Only context entity	77.1
Only context entity concept	78.3
w/o trimming	78.5
Trimming(only context)	77.9
KARN	**79.4**

Table 7. Example in Commonsense for case study.

CSQA Question:

Where would you go to get some pamphlets if you want to own them?

Question entity description:

pamphlets: A pamphlet is an unbound book.

CSQA Answer:

(A) bookstore (B) drawer (C) health department (D) mail box
(E) library

bookstore: Bookselling is the commercial trading of books which is the retail and distribution end of the publishing process.

library: A library is a collection of materials, books or media that are accessible for use and not just for display purposes.

external knowledge. It may be because using external knowledge alone affects the relevance score of the node, resulting in a decline in model performance.

Edge Weight Calculation. As shown in Table 6, the model performance decreases by 2.9% when there are no weights of edges in GNN. And when the edges have weights, the control groups' performance is improved. Additionally, we can observe that the model performance improves by 1.8% when using external knowledge of the node concept features compared to no weights, proving the improved effect of external knowledge on question-answer reasoning. The final result shows that fusing contextual nodes and concepts into the edge weight calculations is reasonable.

4.6 Case Analysis

From Table 7, we can clearly see that after additionally adding the external knowledge base, we can extract a very important keyword from the description of the question entity and the answer entity: book. In the knowledge graph, we obtain two paths from "pamphlets" to "bookstore" and "library". When scoring node relevance, “book” will be given a high score. When GNN updates the node features on these two paths, it will take the conceptual description of the node as a significant factor. At this time, the node “library” path is obviously far away from the context. Finally, the node “bookstore” is selected after calculation.. Finally, the node “bookstore” is selected after calculation.

This case simply illustrates that we fuse context and external knowledge to prune the knowledge graph. And these two kinds of knowledge are applied to the weight calculation of the relation, so as to enhance the node features and get the correct answer.

5 Conclusions

In this paper, we use the LM+KG model to solve the question answering problem. We fuse contextual knowledge with external knowledge, compute relevance scores for entities and questions, and tailor knowledge graphs based on the scores. GNN adds edge weight calculation in the calculation of node features, and takes context and external knowledge as factors affecting edge weights, effectively enhancing node features. By augmenting node features with knowledge, the reasoning performance of the model is effectively improved. Experiments show that our method performs well on two public datasets and one domain-specific (medical) dataset compared to previous LM+KG and LM methods.

Acknowledgements. This work is supported by grant from the National Natural Science Foundation of China (No. 62076048), the Science and Technology Innovation Foundation of Dalian (2020JJ26GX035).

References

1. Chen, J., Hou, H., Gao, J., Ji, Y., Bai, T.: RGCN: recurrent graph convolutional networks for target-dependent sentiment analysis. In: Douligeris, C., Karagiannis, D., Apostolou, D. (eds.) KSEM 2019, Part I. LNCS (LNAI), vol. 11775, pp. 667–675. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29551-6_59
2. Clark, P., et al.: From ‘F’ to ‘A’ on the NY regents science exams: an overview of the aristo project. AI Mag. **41**(4), 39–53 (2020)
3. Feng, Y., Chen, X., Lin, B.Y., Wang, P., Yan, J., Ren, X.: Scalable multi-hop relational reasoning for knowledge-aware question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1295–1309 (2020)
4. Ishiwatari, T., Yasuda, Y., Miyazaki, T., Goto, J.: Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7360–7370 (2020)
5. Jin, D., Pan, E., Oufattole, N., Weng, W.H., Fang, H., Szolovits, P.: What disease does this patient have? A large-scale open domain question answering dataset from medical exams. Appl. Sci. **11**(14), 6421 (2021)

6. Kenton, J.D.M.W.C., Toutanova, L.K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
7. Khashabi, D., et al.: UnifiedQA: crossing format boundaries with a single QA system. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1896–1907 (2020)
8. Lee, J., et al.: BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining, vol. 36, pp. 1234–1240. Oxford University Press, Oxford (2020)
9. Lin, B.Y., Chen, X., Chen, J., Ren, X.: KagNet: knowledge-aware graph networks for commonsense reasoning. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2829–2839 (2019)
10. Liu, F., Shareghi, E., Meng, Z., Basaldella, M., Collier, N.: Self-alignment pretraining for biomedical entity representations. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4228–4238 (2021)
11. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
12. Lv, S., et al.: Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8449–8456 (2020)
13. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? A new dataset for open book question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2381–2391 (2018)
14. Mihaylov, T., Frank, A.: Knowledgeable reader: enhancing cloze-style reading comprehension with external commonsense knowledge. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 821–832 (2018)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learning Res. **21**(1), 5485–5551 (2020)
16. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Speer, R., Chin, J., Havasi, C.C.: 5.5: an open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (December 2016), pp. 4444–4451 (2016)
18. Talmor, A., Herzig, J., Lourie, N., Berant, J.: CommonsenseQA: a question answering challenge targeting commonsense knowledge. In: Proceedings of NAACL-HLT, pp. 4149–4158 (2019)
19. Wang, X., Kapanipathi, P., et al.: Improving natural language inference using external knowledge in the science questions domain. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7208–7215 (2019)
20. Xu, Y., Zhu, C., Xu, R., Liu, Y., Zeng, M., Huang, X.: Fusing context into knowledge graph for commonsense question answering. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1201–1207 (2021)
21. Yan, J., et al.: Learning contextualized knowledge structures for commonsense reasoning. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 4038–4051 (2021)

22. Yasunaga, M., Ren, H., Bosselut, A., Liang, P., Leskovec, J.: QA-GNN: reasoning with language models and knowledge graphs for question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 535–546 (2021)
23. Zhang, X., et al.: GreaseLM: graph reasoning enhanced language models. In: International Conference on Learning Representations (2022)

Oral: Large Language Models

Creative Destruction: Can Language Models Interpret Oxymorons?

Fan Xu, Ziyun Zhu, and Xiaojun Wan(✉)

Wangxuan Institute of Computer Technology, Peking University, Beijing, China
{xufan2000,wanxiaojun}@pku.edu.cn, zhuziyun345@stu.pku.edu.cn

Abstract. Oxymoron is a figurative language which combines seemingly contradictory words in a short phrase. It is used to create an impression, enhance a concept or entertain the readers. In this work, we propose a novel task named oxymoron interpretation, which requires destructing the contradiction and creating new senses to make the phrase comprehensible and harmonious with the context. We construct a dataset, Oxymoron with Context-Based Interpretation(OCBI) by gathering data from the internet and manually annotating it. Furthermore, some prevalent models and methods, including definition generation, GPT-2, T5, ChatGPT, LLaMA and Alpaca, are tested to explicitly generate interpretation. According to automatic and manual evaluation results, models except ChatGPT still struggle to generate plain and context-dependent interpretation, which brings us some insights of the capability gap between models. Our work reveals that oxymoron interpretation is a challenging task requiring common sense and contextual association ability. It can be used to assess how well language models comprehend figurative language. The OCBI dataset is publicly available (https://github.com/pku0xff/oxymoron_interpretation).

Keywords: Oxymoron · Dataset · Language model · Figurative language

1 Introduction

Oxymoron is a figurative language that contains contradictory words in a short phrase or even in a single word. There are three kinds of oxymoron [5]: Objective oxymoron has obvious contradiction; subjective oxymoron heavily depends on personal opinion; punning oxymoron exists because some words have multiple meanings. Among the three types, objective oxymoron is our focus in this study as it is the most classic and typical one. Oxymoron has been widely used in literary works to create an impression, enhance a concept or just entertain the reader. For example, the play Romeo and Juliet contains oxymorons like "loving hate", "sweet sorrow" and "sick health". However, it may be difficult to understand oxymoron for entry-level language learners.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 645–656, 2023.
https://doi.org/10.1007/978-3-031-44693-1_50

Context is necessary to make oxymoron comprehensible [5]. For example, in context "*Getting frustrated at drawing a blank for directing a film, after Sadhana he returned home to Varanasi.*", drawing a blank means all efforts are in vain. While in another context "*If you're drawing a blank, we'll give you a hint :'Sign Boy.'*", it means being unable to recall a memory. We verify the effect of context in our experiments using ChatGPT and find that the generated explanation of oxymoron is usually much more concrete and easy to understand with a given context. However, the vital sense of interpretation often comes from external knowledge not presented in context. Oxymoron interpretation is kind of destructive creation requiring common sense.

In this work, we propose the **oxymoron interpretation** task and build a benchmark dataset named **Oxymoron with Context-Based Interpretation(OCBI)** for evaluating the task. We describe the construction process and analysis of the OCBI dataset in Sect. 3. We further investigate into some prevalent methods on the oxymoron interpretation task. First, we try applying two definition generation models to it. Then we compare two pre-trained models, GPT-2 and T5, by fine-tuning or prompt-tuning them on our dataset. Three popular large language models(LLMs), ChatGPT, LLaMA-7B and Alpaca-7B are also applied to the task. Evaluation results indicate that oxymoron interpretation is much more difficult than definition generation. GPT-2, T5 and LLaMA still struggle to interpret oxymoron while ChatGPT and Alpaca can often understand oxymorons. ChatGPT can even generate highly context-dependent and detailed interpretation.

In short, our contributions can be summarized into following points:

1. We propose the new task of oxymoron interpretation, which can be used to test the ability of language models to understand figurative language. Automatic oxymoron interpretation is also a help for language learners.
2. We build and release the OCBI dataset. It contains 288 oxymorons with 1217 contexts in total and each entry contains two high-quality human-written interpretations.
3. We investigate a series of methods to interpret objective oxymorons and find that small pretrained models like GPT-2 and T5 fail on this task while ChatGPT achieves good performance. The gaps between models help research on model abilities, especially common sense.

2 Related Work

2.1 Figurative Language Processing

Although there has been no study about oxymoron interpretation as far as we know, some studies focused on oxymoron detection. Cho et al. [3] proposed the method of constructing a set of offset vectors and using them to compare the word pairs and detect antonyms in the statement. La Pietra and Masini [13] detected a list of oxymorons by extracting antonym pairs from a corpus. As for interpretation of figurative language, He et al. [8] proposed a method to interpret simile. Chakrabarty et al. [2] used COMET [1] as knowledge enhancement in idiom and simile interpretation. Qiang et al. [21] proposed a knowledge-enhanced seq2seq model to interpret Chinese idioms.

2.2 Definition Generation (DG)

Noraset et al. [19] first defined the task of definition generation. Later, many subsequent works used the context of a given word to deal with the task ([7,15,17]). Ishiwatari et al. [10] used local and global context to define an unknown phrase. Further, some studies used pre-trained language models to solve the problem ([12,24,28]). Huang et al. [9] fine-tuned T5 [23] for the task and proposed a method to re-rank the definitions with specificity. Oxymoron interpretation is similar to definition generation task, but it is more difficult for we need to destruct the contradiction and create new senses.

3 Task and Dataset

Table 1. Statistics of dataset.

Split	#Phrases	#Entries	Context len.
Train	139	600	16.03
Valid	76	316	16.20
Test	73	301	16.30

Table 2. The classification result of oxymoron phrases.

Type	Number	Proportion
Objective	1212	71.84%
Subjective	383	22.70%
Punning	26	1.54%
Unknown	66	3.92%

Task Definition. The oxymoron interpretation task is to explicitly generate interpretation text of an oxymoron given its context. We expect the interpretation to be literal, straightaway, detailed, and consistent with the context. The most important and difficult point is to destruct contradiction and create new senses from the seemingly opposite concepts and their context.

Data Collection. We construct the OCBI dataset from a list of oxymoron phrases rather than extract them from literary works. Actually, oxymoron is not used as frequently as some other figurative languages such as simile, metaphor or sarcasm. It is hard to generate as well because only certain words used in certain contexts can make sense. We first obtain a list containing 1687 oxymoron phrases from a website[1]. For each oxymoron in this list, we search sentences that contain the oxymoron. Most sentences are crawled from online dictionaries. Finally, we get 9166 sentences covering 504 oxymoron phrases in total.

[1] http://www.oxymoronlist.com/.

Data Preprocessing. After collecting oxymoron phrases and their contexts, we filter the sentences by length and only preserve sentences two more words longer than the oxymoron phrase. To make contexts as variable as possible, we use K-means [16] algorithm to aggregate similar sentences for each oxymoron and filter out redundant sentences.

Oxymoron Classification. We hire human annotators to distinguish objective, subjective and punning oxymorons. Each oxymoron with context is annotated by three people and the final type is decided by the majority. If an oxymoron is classified into three different types by three annotators, we identify its type as "unknown". If an annotator really do not know how to classify an oxymoron, he or she can also mark it as "unknown". We calculate the inter-rater agreement metric, Fleiss' Kappa [6], and get a result of 0.4334, indicating there's a moderate agreement. The classification result is shown in Table 2. Most oxymorons in our data are objective ones, the most classic and typical type. In subsequent experiments, we focus on this type only.

Language Style Analysis. The OCBI dataset encompasses diverse language styles. We use ChatGPT to classify the language style of the context sentences in the dataset according to two criteria: formality and tone. The questions are "*Is this sentence formal, neutral or informal?*" and "*Is this sentence in a serious, neutral or casual tone?*". We concatenate context sentences and the two questions respectively as the prompts. For formality, the distribution of categories is as follows: formal (31.6%), neutral (47.3%), informal (21.0%) and unrecognizable (0.2%). In terms of tone, the categories are distributed in the following manner: serious (37.5%), neutral (40.2%), causal (20.2%) and unrecognizable (2.1%).

Interpretation Writing. To build a high-quality dataset for oxymoron interpretation, we select part of the oxymorons randomly and hire postgraduates majoring in English to write two interpretations for each (*oxymoron*, *context*) pair. The average length of the human-written interpretations is 11.05. Human evaluation results (Table 4) confirm the human-written interpretations are of high quality. Statistical information of the final dataset is shown in Table 1. It contains 288 oxymoron phrases and 1217 entries in contexts in total, with each oxymoron entry having two interpretations.

4 Experiments

We conduct a series of experiments to explore whether definition generation models can accomplish the oxymoron interpretation task, and how pre-trained models including GPT-2, T5, ChatGPT, LLaMA and Alpaca perform on this task.

4.1 Definition Generation Methods

Definition Modelling for Appropriate Specificity (DMAS) [9]. We apply the DMAS model to oxymoron interpretation task and use WordNet dataset ([10,18,19]) to train the model. We set the proportion of generation likelihood, over-specificity score and under-specificity score to 0.15, 0.6 and 0.25 respectively as suggested. Hyper-parameters are also kept the same.

Learning to Describe Unknown Phrases with Local and Global Contexts (LOG-CaD) [10]. Considering many oxymoron phrases are uncommon, we test another definition generation model which specifies in interpreting unknown phrases. We use Oxford dataset [7] to train the model. We set the training epoch to 20 and batch size to 32, and leave other settings unchanged.

4.2 Parameter Tuning

Prompt-Tuning (PT) [14]. As the training set in OCBI is not large, prompt-tuning may be an appropriate method that only requires to training a few parameters. We try two pre-trained models, GPT-2(124M)[2] [22] and T5(220M)[3] [23]. The trainable prompt is initialized with text "Use the context and knowledge to interpret the phrase". For GPT-2, the format of input is `PROMPT` $\oplus$ `''The phrase PHRASE has a context: CONTEXT. Therefore, the phrase means:''`. For T5, it is `PROMPT` $\oplus$ `''Phrase: PHRASE. Context: CONTEXT''`.[4] We use Adam [11] optimizer and set the learning rate to 0.1 for both models. GPT-2 prompt is trained for 10 epochs with a batch size of 1 and updated every 4 steps. T5 prompt is trained for 20 epochs with a batch size of 4.

Fine-Tuning (FT). We also try fine-tuning GPT-2 and T5 with a learning rate of 5e−5 for 1 and 5 epochs respectively. The input format is the same as prompt-tuning experiments.

4.3 Prompt

ChatGPT. ChatGPT has achieved good performance on many NLP tasks and quickly becomes the most popular language model recently. We test ChatGPT (`gpt-3.5-turbo`) with simple prompts on our dataset and examine the necessity of the context. We design two prompts: (1) `[CONTEXT] What does [PHRASE] mean in this sentence?` (2) `What does [PHRASE] mean?`

LLaMA [26] *and Alpaca* [25]. LLaMA-7B and Alpaca-7B, though much smaller than ChatGPT, are competent on many NLP tasks. For these two models, the prompt is "`[CONTEXT] In this sentence, [PHRASE] means`". The content completed by language models is taken as interpretation.

[2] https://huggingface.co/gpt2.

[3] https://huggingface.co/t5-base.

[4] Here the symbol $\oplus$ means concatenation.

4.4 Evaluation

We use BLEU [20], NIST [4] and BERTScore-recall(BS-R) [27] as automatic evaluation metrics. As generated interpretation can be more detailed than the references, we use recall rather than F1 when calculating BERTScore. To accommodate diversity, we take the maximum score between each candidate sentence and its reference sentences, rather than an average. We also conduct human evaluation. We randomly sample 50 entries from test set and hire three annotators to finish the evaluation. To make the evaluation as fair as possible, we provide a detailed scoring criterion (Table 3). The interpretation whose average score exceeds 3.5 are defined as acceptable.

Table 3. Detailed scoring criteria of human evaluation.

Score	Criterion
1	Interpretation not related to the oxymoron, giving totally wrong information, or just repeating the oxymoron
2	Interpretation related to the oxymoron and context but failing to eliminate the contradiction
3	Incomplete interpretation, or interpretation not consistent with the context
4	Correct interpretation missing small details
5	Correct and detailed interpretation consistent with the context

5 Results and Analysis

Automatic and manual evaluation results are shown in Tables 4. The LOG-CaD model performs poorly in automatic evaluation, so we do not evaluate its prediction manually any more. In general, the tested methods can be divided into three grades based on their performance: (1) definition generation methods, GPT-2, T5 and LLaMA still struggle to interpret most oxymoron; (2) Alpaca can sometimes generate coarse-grained but reasonable interpretation; (3) ChatGPT can offer fine-grained and highly context-sensitive interpretation with simple prompts. We dig into the gaps between these methods by observing how oxymoron words and context words are mentioned in predictions. The proportion situation is illustrated in Fig. 1. The values are calculated by dividing the length of oxymoron or context by the number of mentioned tokens.

Interpreting Oxymoron Requires Destructing Contradiction. We detect whether oxymoron words, their variants or synonyms are mentioned in the generated interpretation and the results are shown in Fig. 1 as yellow bars. ChatGPT often repeats the oxymoron phrase in its interpretation (e.g., it generates "[OXYMORON] means ...") so its proportion is close to 1. GPT-2 and T5 models

Table 4. Evaluation results.

Method		Auto			Manual	
		BLEU	NIST	BS-R	Mean	Acc.
DG	DMAS	0.0683	0.1172	0.4997	2.17	0.08
	LOG-CaD	0.0159	0.1640	0.3874	–	–
GPT-2	PT	0.1048	0.6799	0.5588	2.11	0.06
	FT	0.0938	0.7010	0.5446	2.20	0.10
T5	PT	0.0634	0.2658	0.5095	2.08	0.16
	FT	0.0867	0.5663	0.5600	2.62	0.20
LLaMA	Prompt	0.0869	0.3529	0.5427	2.68	0.28
Alpaca		**0.1055**	0.6314	0.5870	3.40	0.52
ChatGPT (w/o context)		0.0724	**0.7337**	**0.6214**	**4.59**	**0.94**
		0.0524	0.5202	0.6026	4.28	0.88
Human		–	–	–	4.19	0.84

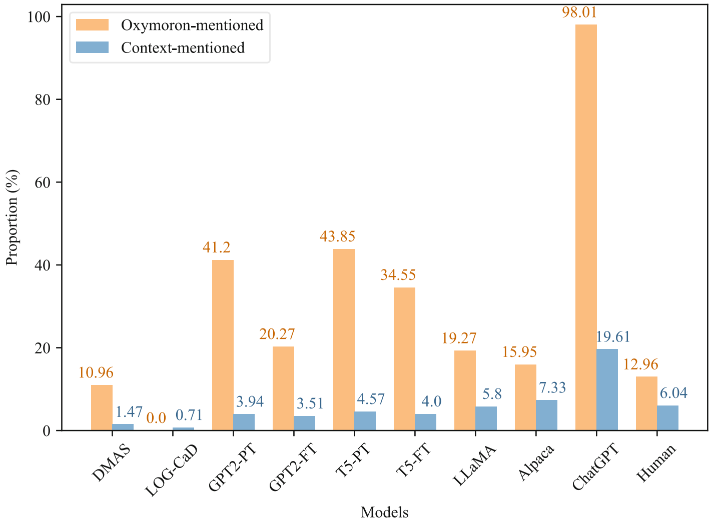


Fig. 1. Oxymoron- and context-mentioned proportion in model predictions. (Color figure online)

achieve higher oxymoron-mentioned proportion than humans but receive low meaning score in manual evaluation. That's because they are still ignorant of the task requirements and tend to repeat the oxymoron phrase as its interpretation. Another common fault is one-sidedness, i.e., focusing on a single part of the oxymoron and totally ignore the contradiction. For example, "realistic fantasy" means an artistic style in its given context, but prompt-tuned T5 interprets it as "a fantasy that is not realistic". These two types of faults also cause the gap between LLaMA and Alpaca. Sometimes the actual meaning can be far away from the literal meaning so the interpretation may not include the original words. For example, "drawing a blank" can be interpreted as "being unable to come up with an idea or solution". A valid interpretation should express the symbolic meaning rather than merely repeat the literal meaning of the oxymoron.

Oxymoron Interpretation Relies on the Context. For models include context sentences in their input, we count how many context words are mentioned in their outputs and the result proportion is shown in Fig. 1 as blue bars. There is an obvious positive correlation between context-mentioned proportion and human evaluation score, indicating the importance of context information. Additionally, we conduct a context-excluded experiment on ChatGPT and verify the importance of context. ChatGPT has a strong ability to understand contextual relationships. When context is missed, there's a sharp drop in NIST metric and the interpretation is sometimes ambiguous.

Oxymoron Interpretation Differs from Definition Generation. It can be inferred from Table 4 that definition generation methods are still deficient in oxymoron interpretation task. Indeed, the DMAS model seldom makes the mistake of repeating oxymoron as its output, and generates more fluent and simple outputs than GPT-2 and T5. This can be attribute to the WordNet dataset used to train it, as oxymoron interpretation is similar to definition generation and the WordNet dataset is much larger than our OCBI dataset. However, different from oxymoron, normal phrases are more combinations of semantically consistent words. When processing the oxymoron interpretation task, the definition generation models tend to generate incomplete or even wrong outputs. This can also confirm that the oxymoron interpretation task is different from and more challenging than definition generation.

6 Discussion

We have to admit that there are some points that may arouse disagreement in the task and dataset. First, everyone may have a different opinion on the definition of oxymoron. Some oxymoron phrases, e.g. *constant change* and *tiny mountain*, are so frequently used and straightforward that we rarely treat them as figurative in our daily life. Second, a small part of context sentences do not contain oxymoron actually because of their syntax. For example, in the context "*tiny mountain village*", "tiny" is used to describe "village". These contexts are included because existing methods to detect oxymoron is still insufficient.

7 Conclusion

In this work, we propose a challenging task, oxymoron interpretation, and the OCBI dataset. We further investigate some definition generation methods and pretrained models on the oxymoron interpretation task. There are significant gaps between models, especially between ChatGPT and other models. Smaller models tend to ignore the contradiction and therefore wrongly focus on a single part in opposition or just repeat the oxymoron. Overall, oxymoron interpretation is a task that requires common sense and contextual understanding ability and can be used to test the language model's ability in these aspects.

A Appendices

A.1 Interpretation Examples

(See Tables 5 and 6).

Table 5. Some **bad** examples of generated interpretations. The oxymoron phrase is "*new cliche*" in context "*Can't you think of a new cliche?*"

Method		Interpretation
DG	DMAS	a familiar term for an old cliché
	LOG-CaD	a `<unk>`
GPT-2	PT	a new cliche that is not new but is not new at all
	FT	a phrase that has a new meaning or context
T5	PT	a cliche that is not a new one
	FT	a new cliche that is used to describe something that is not new or that is not old
LLaMA		a new cliché
Alpaca		a phrase or expression that has not been used before

Table 6. Some **good** examples of generated interpretations. The oxymoron phrase is "*diminished growth*" in context "*If the advanced-country scenario is right, many developing countries will lose out and experience sharply diminished growth.*"

Method		Interpretation
DMAS		a decrease in rate of growth
GPT-2	PT	a decrease in the growth rate of the economy
	FT	a situation in which the economy is growing at a slow rate
T5	PT	a country that has a high growth rate but is not able to sustain it
	FT	a decline in growth
LLaMA & Alpaca		the growth rate is decreasing
ChatGPT (w/o context)		"diminishing growth" means a decrease in the rate of economic growth or a slowdown in the pace of development
		Diminishing growth refers to a situation where the rate of growth of a particular variable, such as economic output or population, slows down over time

References

1. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: commonsense transformers for automatic knowledge graph construction. In: Proceedings of the 7th Annual Meeting of the Association for Computational Linguistics, pp. 4762–4779. Association for Computational Linguistics, Florence, Italy, July 2019. https://doi.org/10.18653/v1/P19-1470, https://aclanthology.org/P19-1470
2. Chakrabarty, T., Choi, Y., Shwartz, V.: It's not rocket science: interpreting figurative language in narratives. Trans. Assoc. Comput. Linguist. **10**, 589–606 (2022). https://doi.org/10.1162/tacl_a_00478, https://aclanthology.org/2022.tacl-1.34
3. Cho, W.I., Kang, W.H., Lee, H.S., Kim, N.S.: Detecting oxymoron in a single statement. In: 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA), pp. 1–5 (2017). https://doi.org/10.1109/ICSDA.2017.8384447
4. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on Human Language Technology Research, pp. 138–145 (2002). https://dl.acm.org/doi/abs/10.5555/1289189.1289273
5. Flayih, M., et al.: A linguistic study of oxymoron. J. Kerbala Univ. **5**(1), 30–40 (2009). https://www.iasj.net/iasj/download/cab38a074e4f8022
6. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378 (1971). https://doi.org/10.1037/h0031619

7. Gadetsky, A., Yakubovskiy, I., Vetrov, D.: Conditional generators of words definitions. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 266–271. Association for Computational Linguistics, Melbourne, Australia, July 2018. https://doi.org/10.18653/v1/P18-2043, https://aclanthology.org/P18-2043
8. He, Q., Cheng, S., Li, Z., Xie, R., Xiao, Y.: Can pre-trained language models interpret similes as smart as human? In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 7875–7887. Association for Computational Linguistics, Dublin, Ireland, May 2022. https://doi.org/10.18653/v1/2022.acl-long.543, https://aclanthology.org/2022.acl-long.543
9. Huang, H., Kajiwara, T., Arase, Y.: Definition modelling for appropriate specificity. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 2499–2509. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, November 2021. https://doi.org/10.18653/v1/2021.emnlp-main.194, https://aclanthology.org/2021.emnlp-main.194
10. Ishiwatari, S., et al.: Learning to describe unknown phrases with local and global contexts. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 3467–3476. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1350, https://aclanthology.org/N19-1350
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980
12. Kong, C., Chen, Y., Zhang, H., Yang, L., Yang, E.: Multitasking framework for unsupervised simple definition generation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5934–5943. Association for Computational Linguistics, Dublin, Ireland, May 2022. https://doi.org/10.18653/v1/2022.acl-long.409, https://aclanthology.org/2022.acl-long.409
13. La Pietra, M., Masini, F.: Oxymorons: a preliminary corpus investigation. In: Proceedings of the Second Workshop on Figurative Language Processing, pp. 176–185. Association for Computational Linguistics, Online, July 2020. https://doi.org/10.18653/v1/2020.figlang-1.24, https://aclanthology.org/2020.figlang-1.24
14. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3045–3059. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, November 2021. https://doi.org/10.18653/v1/2021.emnlp-main.243, https://aclanthology.org/2021.emnlp-main.243
15. Li, J., Bao, Y., Huang, S., Dai, X., Chen, J.: Explicit semantic decomposition for definition generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 708–717. Association for Computational Linguistics, Online, July 2020.https://doi.org/10.18653/v1/2020.acl-main.65, https://aclanthology.org/2020.acl-main.65

16. MacQueen, J.: Classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967). https://www.cs.cmu.edu/bhiksha/courses/mlsp.fall2010/class14/macqueen.pdf
17. Mickus, T., Paperno, D., Constant, M.: Mark my word: a sequence-to-sequence approach to definition modeling. In: Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing, pp. 1–11. Linköping University Electronic Press, Turku, Finland, September 2019. https://aclanthology.org/W19-6201
18. Miller, G.A.: WordNet: a lexical database for English. In: Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, 23–26 February 1992. https://aclanthology.org/H92-1116
19. Noraset, T., Liang, C., Birnbaum, L., Downey, D.: Definition modeling: learning to define word embeddings in natural language. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, February 2017. https://doi.org/10.1609/aaai.v31i1.10996, https://ojs.aaai.org/index.php/AAAI/article/view/10996
20. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, July 2002. https://doi.org/10.3115/1073083.1073135, https://aclanthology.org/P02-1040
21. Qiang, J., et al.: Chinese idiom paraphrasing. arXiv preprint arXiv:2204.07555 (2022). https://arxiv.org/abs/2204.07555
22. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019). https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
23. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html
24. Reid, M., Marrese-Taylor, E., Matsuo, Y.: VCDM: leveraging variational bi-encoding and deep contextualized word representations for improved definition modeling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6331–6344. Association for Computational Linguistics, Online, November 2020. https://doi.org/10.18653/v1/2020.emnlp-main.513, https://aclanthology.org/2020.emnlp-main.513
25. Taori, R., et al.: Stanford alpaca: an instruction-following llama model (2023). https://github.com/tatsu-lab/stanford_alpaca
26. Touvron, H., et al.: LLaMA: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
27. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: evaluating text generation with BERT. CoRR abs/1904.09675 (2019). http://arxiv.org/abs/1904.09675
28. Zheng, H., et al.: Decompose, fuse and generate: a formation-informed method for Chinese definition generation. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5524–5531. Association for Computational Linguistics, Online, June 2021. https://doi.org/10.18653/v1/2021.naacl-main.437, https://aclanthology.org/2021.naacl-main.437

Global Prompt Cell: A Portable Control Module for Effective Prompt Tuning

Chi Liu, Haochun Wang, Nuwa Xi, Sendong Zhao(✉), and Bing Qin

Harbin Institute of Technology, Harbin, China
{cliu,hcwang,nwxi,sdzhao,bqin}@ir.hit.edu.cn

Abstract. As a novel approach to tuning pre-trained models, prompt tuning involves freezing the parameters in downstream tasks while inserting trainable embeddings into inputs in the first layer. However, previous methods have mainly focused on the initialization of prompt embeddings. The strategy of training and utilizing prompt embeddings in a reasonable way has become a limiting factor in the effectiveness of prompt tuning. To address this issue, we introduce the Global Prompt Cell (GPC), a portable control module for prompt tuning that selectively preserves prompt information across all encoder layers. Our experimental results demonstrate a 5.8% improvement on SuperGLUE datasets compared to vanilla prompt tuning.

1 Introduction

Prompt-based methods can be classified into two categories: discrete prompt tuning [1,15] and continuous prompt tuning [8]. Discrete prompt tuning transforms the task into a "fill-in-the-blank" format and then utilizes a pre-trained language model to predict the answer, which operates similarly to the masked language model (MLM) [2]. In subsequent research, continuous prompt tuning [8] introduced soft prompts (i.e., prompt embeddings) to replace manual templates, which consist of special tokens with adjustable embeddings. We refer to continuous prompt tuning as "prompt tuning" for simplicity.

Vanilla prompt tuning, which concatenates prompt embeddings with input tokens in the first layer and updates only the parameters of the prompt embeddings during the training phase, has several limitations [8,21]. Firstly, since the effectiveness of prompt embeddings is highly related to the length, it is necessary to use prompt embeddings with hundreds of tokens in length to achieve better downstream task performance, as suggested by [8] and [21]. However, long prompt embeddings also reduce the possible length of input text. In addition, prompt learning requires a longer time to converge compared to full fine-tuning [22], and its effectiveness still has significant room for improvement.

These drawbacks are due to the traditional approach of inserting prompt embeddings into the input layer and concatenating them with token embeddings for model input in prompt learning. However, prompt embeddings have significant differences compared to token embeddings. Firstly, prompt embeddings

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 657–668, 2023.
https://doi.org/10.1007/978-3-031-44693-1_51

have not undergone pre-training and require more optimization steps compared to token embeddings. Secondly, prompt embeddings do not have semantic information but serve as task-specific vectors to guide the model for downstream tasks. Finally, in pre-trained models with multiple layers, prompt learning freezes the parameters during training, updating only the bottom-level prompt embeddings, which can cause long-distance backpropagation to result in vanishing gradients and slow convergence [18]. These reasons indicate the need to design better training and utilization methods for prompt embeddings, instead of using the same training method as token embeddings.

The RNNs contain a hidden unit that can preserve important information during sequence iterations and guide the model to output results [19]. RNNs can be unfolded into a long sequence, and through the hidden unit, the model can selectively integrate information from different times from the beginning to the end. The input at the initial time can guide the judgment at the final time, and this approach is beneficial in alleviating the problem of vanishing gradients, reducing forgetting, and speeding up convergence.

Prompt embeddings suffer from information loss when passing through each layer of transformers. Although residual connection modules exist within each layer, these modules operate on the entire sequence, including the token embeddings, rather than the prompt embeddings alone. Therefore, due to the reasons mentioned earlier, we need to design a dynamic information fusion mechanism specifically for the part corresponding to the prompt embeddings. Inspired by RNNs, we propose the Global Prompt Cell (GPC) to address the aforementioned issues.

The Global Prompt Cell (GPC) consists of two units: the remembering unit and the forgetting unit. Remembering unit should be applied to the prompt embeddings before passing through the transformer layer because information loss occurs after passing through the layers. Therefore, we use remembering unit to selectively remember certain information. On the other hand, forgetting unit should be applied to the prompt embeddings after passing through the transformer layers. The forgetting unit can selectively forget some information in the latest prompt embeddings to fuse with the previous prompt embeddings.

By utilizing the same remembering unit and forgetting unit for every model layer, our approach gathers information regarding prompt embeddings from all layers, enabling it to guide the model to achieve better downstream task performance. As a result, prompt embeddings are no longer simply a vector concatenated with token embeddings, but also a control module that aids PTMs in making better decisions. Moreover, since GPC only acts on the prompt embeddings updates between layers, it can be considered a plug-in module and requires only a small number of additional parameters. Lastly, we eliminate the verbalizer in vanilla prompt tuning and use classification heads for downstream tasks, reducing the difficulty of selecting optimal verbalizers for various tasks.

To summarize, our contributions are three-fold:

1. We propose a new training and utilization method for prompt embeddings called the Global Prompt Cell (GPC). To our knowledge, GPC is

the first method that specifically aims to improve the training and utilization of prompt embeddings, reduce forgetting during training, and ultimately enhance downstream task performance. Experiments prove its effectiveness and show its validity in architecture with ablation study.
2. Our method simplifies prompt tuning by discarding the verbalizer, which further reduces the time and computing consumption to select the optimal verbalizer, and achieves even better results.
3. Our method can be viewed as an easy-to-implement plug-in module with only a few additional parameters, which makes GPC both model-agnostic and task-agnostic.

2 Related Work

2.1 Pre-trained Language Model

Recently substantial works have shown that pre-trained models (PTMs) can learn universal language representations through pre-trained tasks on large corpora, which are beneficial for downstream NLP tasks and can avoid training a new model from the beginning [14].

A representative application of PTMs is using encoder-based models for classification tasks [23]. Encoder-based models are composed of multi-layer transformer encoders, which include a self-attention module and a multi-layer perceptron (MLP), and are optimized for performance through measures such as intra-layer residual connections. BERT is a classic encoder-based model commonly used for classification tasks [2]. In BERT, each input sentence is concatenated with a [CLS] token at the beginning. After passing through the encoder of each layer, the [CLS] token is utilized as a classification indicator to produce the final result.

2.2 Prompt Tuning

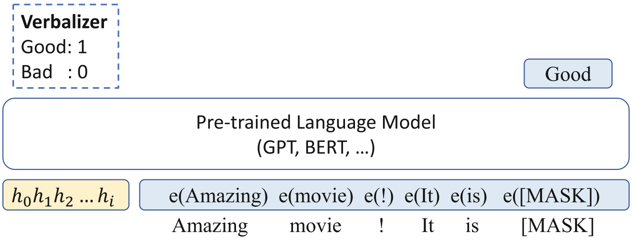

Verbalizer
Good: 1
Bad : 0
Good
Pre-trained Language Model
(GPT, BERT, ...)
$h_0 h_1 h_2 \dots h_i$
e(Amazing) e(movie) e(!) e(It) e(is) e([MASK])
Amazing movie ! It is [MASK]

Fig. 1. Prompt-tuning model.

GPT-3 [1] has revolutionized downstream tasks by transforming them into generation tasks through the addition of prompt-like hints. This allows the model to generate results directly in few-shot or zero-shot learning scenarios. Prompt consists of a template for transforming the input text and a verbalizer for matching the generated words to the actual task labels. It can be designed manually by domain experts [16,17] or automatically [3,20], but at the expense of low explainability [20] (Fig. 1).

Prompt tuning methods with prompt embeddings have been explored in recent studies [5,11,13,27]. Prompt embeddings are trainable embeddings rather than natural language tokens. Vanilla prompt tuning, proposed by [8], concatenates the prompt embeddings with the token embeddings in the first encoder layer.

To use prompt tuning for downstream tasks, we first encode a sequence of discrete input tokens $X = x_0, x_1, ..., x_n$ into embeddings $X_e = e(x_0), e(x_1), ..., e(x_n)$ using a pre-trained language model M. We then obtain the prompt embeddings $P = p_1, p_2, ..., p_n$, where p_i is the prompt embedding for $i \in n$, and concatenate the prompt embeddings and token embeddings to form the complete input $I = P; X_e$. The model output embedding of the [MASK] token is then fed into a classifier to predict the target token, and the verbalizer is used to obtain the actual label.

Current research on prompt tuning architecture retains the verbalizer as a "necessary" component, despite its high computational cost, both in terms of time and computing resources, even when constructed automatically. Despite the improvements observed in downstream tasks, the results of prompt tuning can still be inconsistent or less-than-ideal.

Prompt tuning is a newly-arising paradigm that requires significantly more training time than fine-tuning to achieve the same performance [21], despite the two paradigms only differing in model inputs. This is due to the fact that although prompt embeddings are fundamentally different from token embeddings in terms of initialization and acquisition, vanilla prompt tuning treats prompt embeddings in a similar manner to prompt-like token vectors and processes them together using the same method.

2.3 Model Degradation and Prompt Forgetting

When training deep artificial neural networks, two problems become increasingly challenging for model optimization. The first problem is vanishing gradients [7], which impedes convergence from the outset, but can be largely mitigated through normalized initialization and intermediate normalization layers. The second problem is the degradation problem [6]: as the network becomes deeper, accuracy saturates and then degrades rapidly, despite not being caused by overfitting, which suggests that not all systems are equally easy to optimize [6]. Skip connections [26] are a common countermeasure for model degradation, as seen in ResNet [6] and Transformer [24], where residual connections are employed. Skip connections serve as a reminder for the model to retain previous information and prevent forgetting.

Prompt tuning is also affected by these problems since it occurs at the lowest layer. No previous studies have addressed the issues of model degradation and prompt forgetting in prompt tuning, and we take these problems into consideration.

3 Method

In this section, we present the architecture of our method, which incorporates the implementation of Global Prompt Cell (GPC) into an encoder-based model. Drawing inspiration from RNN models, we utilize a method to effectively store previous states of prompt embeddings in various layers, allowing the model to zoom out and concentrate on the prompt from a wider perspective.

3.1 GPC Between the Encoders

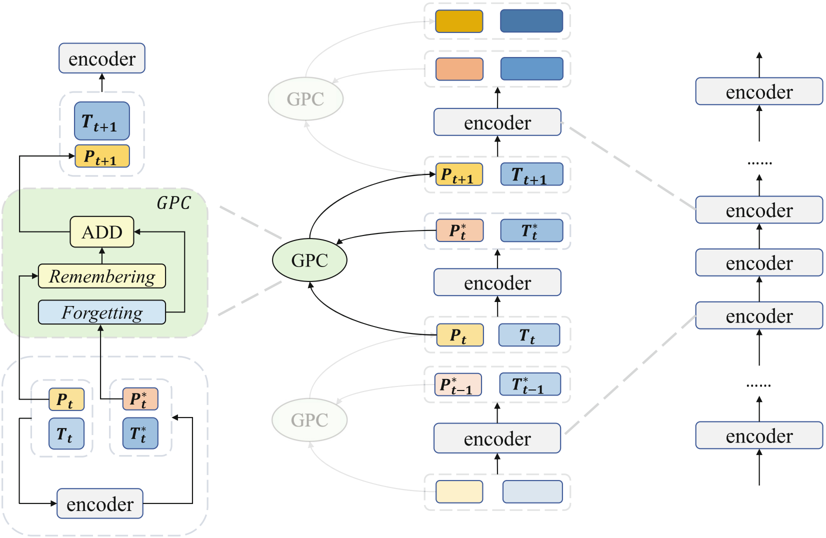


Fig. 2. Global Prompt Cell Model (best viewed in color).

In the Transformer model, residual connections are employed to mitigate the problem of degraded training accuracy. [24]. Building on this idea, GPC employs a more advanced connection mechanism around each pair of adjacent encoders. Specifically, GPC operates exclusively on the prompt portion of each vector. We use P to represent prompt embeddings and T to represent token embeddings, with the two concatenated to form the complete input or output for prompt tuning. As illustrated in Fig. 2, P_t represents the prompt input of the tth encoder, while P_t represents the prompt output of the tth encoder. GPC takes the input and output of the tth encoder, P_t and P_t^*, to generate the prompt input of the

$(t+1)$th encoder E_t, P_{t+1}. Similarly, T_t represents the text input embeddings of the tth encoder, while T_t represents the text output embeddings of the tth encoder, derived from T_t. Since GPC does not affect text embeddings, T_{t+1} is identical to T_t. The complete embeddings are updated as follows.

$$\{P_t^*; T_t^*\} = E_t(\{P_t; T_t\}) \tag{1}$$

The prompt embeddings are updated as below.

$$P_{t+1} = GPC(P_t^*, P_t) \tag{2}$$

3.2 Inside the GPC

Figure 2 provides a detailed view of the internal structure of GPC. The cell comprises two units: the remembering unit and the forgetting unit. Each unit contains a single-layer feed-forward neural network that is shared across all encoder layers. Similar to ResNet, the remembering unit utilizes the prompt embedding before it is fed to the encoder, in order to emphasize and make use of the original information. In contrast, the forgetting unit operates on the output of the encoder to reduce its influence and prevent extreme outcomes. The cell then combines the outputs of both units to produce the final result. Therefore, between every pair of adjacent encoders, GPC considers both the original and encoded prompt embeddings, enabling the model to retain past information and encouraging it to forget some of the current state, thereby striking a balance between the past and the present. The following equation demonstrates how GPC processes prompt embeddings:

$$P_{t+1} = \theta(W_F P_t^* + W_R P_t) \tag{3}$$

where W_F represents the weights of the forgetting unit, while W_R represents the weights of the remembering unit, where θ is the activation function.

3.3 Classification Head

Traditional discrete prompt tuning is mainly used for few-shot or zero-shot tasks, so verbalizers are needed to align with pre-training cloze tasks, in order to narrow the gap between pre-training tasks and downstream tasks, and improve downstream task performance. In contrast, the continuous prompt tuning used in our paper aims to reduce the amount of parameter training and storage during fine-tuning. It generally uses the entire dataset instead of few-shot scenarios, and selecting an appropriate verbalizer manually requires a lot of training resources.

In order to simplify the model and reduce manual intervention as in [8] and [11], we replace the original verbalizers with a classification head that receives outputs from the final layer of the model. We use a randomly initialized classification head to predict the label from the output of the [CLS] token.

4 Experiment

Table 1. Statistics of SuperGLUE datasets.

Corpus	Train	Dev	Test	Task	Metrics
BoolQ	9427	3270	3245	Question Answering	accuracy
CB	250	57	250	Natural Language Inference	accuracy
COPA	400	100	500	Question Answering	accuracy
RTE	2500	278	300	Natural Language Inference	accuracy
WiC	6000	638	1400	World Sense Disambiguation	accuracy
WSC	554	104	146	Co-reference Resolution	accuracy

Table 2. Results on SuperGLUE development set. PT: Prompt tuning [8]; Prompt-only: Prompt tuning with no verbalizer; GPC: Prompt tuning with Global Prompt Cell; **bold**: the best performance.

	BoolQ			RTE		
	PT	Prompt-only	GPC	PT	Prompt-only	GPC
BERT	67.2	62.8	**67.9**	53.5	54.5	**61.0**
RoBERTa	62.3	62.4	**63.5**	58.8	54.2	**59.4**
	CB			COPA		
	PT	Prompt-only	GPC	PT	Prompt-only	GPC
BERT	80.4	71.4	**82.1**	55.0	58.0	**67.0**
RoBERTa	71.4	69.6	**73.2**	63.0	62.0	**66.0**
	WiC			WSC		
	PT	Prompt-only	GPC	PT	Prompt-only	GPC
BERT	63.0	56.4	**66.9**	64.4	64.4	**65.4**
RoBERTa	56.9	54.7	**69.6**	64.4	63.5	**65.4**

4.1 Datasets and Metrics

We evaluate on SuperGLUE [25]. SuperGLUE is a new benchmark styled after GLUE with a new set of more difficult language understanding tasks. Because our task mainly involves classification tasks, while MultiRC and ReCORD belong to QA tasks, and these two tasks are difficult to handle using prompt learning, as reported by [9]. Existing prompt learning methods for these two tasks suffer from significant fluctuations and are difficult to converge. Moreover, the effectiveness of our method in classification tasks has been demonstrated through the other six tasks. Therefore, we selected the other six classification tasks for our experiments. We choose the classification tasks and co-reference resolution tasks, including BoolQ, RTE, CB, COPA, WiC and WSC. Statistics of the selected datasets are in Table 1. We use accuracy as our evaluation metric.

4.2 Experiment Settings

Models. We employ BERT [2] and RoBERTa [12] for our model, both of which are based on transformer encoders and are typically used for classification tasks.

Prompt Length. The length of prompt embeddings has a significant impact on model performance. According to [10], different tasks require prompt embeddings of varying lengths to achieve optimal results. Typically, simpler tasks require shorter prompts than more complex ones [10]. We experimented with prompt lengths of 16, 32, and 64 and selected the most effective ones among them.

Prompt Initialization. Prompt embeddings can be initialized in various ways, such as random initialization or utilizing concrete token embedding. Based on the approach described in [4], we use random initialization in our method.

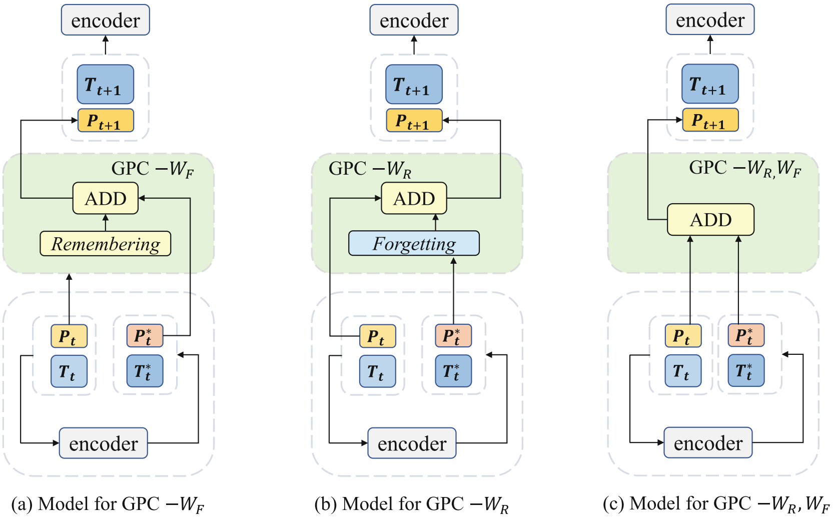
Training Method. During the training phase, we freeze the original parameters of the PTMs and only update the Global Prompt Cell, specifically the corresponding weight matrices W_R and W_F. Therefore, we only need to store a small number of parameters instead of the parameters of the entire PTM for each downstream task.

4.3 Main Results

To assess the effectiveness of Global Prompt Cell, we investigate (i) whether replacing the verbalizer with a classification head causes performance degradation in prompt tuning, and (ii) whether Global Prompt Cell can outperform prompt tuning. (iii) whether Global Prompt Cell combined with a classification head still yield better results.

We carry out prompt-only experiments, where we substitute the verbalizer with a classification head and refrain from using GPC. Table 2 demonstrates that the performance of prompt-only models decreases, particularly on the CB dataset, with both BERT and RoBERTa models experiencing over 10% decrease.

For (ii) and (iii), we perform experiments with Global Prompt Cell. As shown in Table 2, GPC significantly improves the effectiveness of prompt tuning using classification heads, surpassing the performance of vanilla prompt tuning. GPC can achieve over 10% performance increase compared to the prompt-only model on both WiC and CB datasets. In comparison with the PT (Prompt-Tuning [8]) model, which requires multiple experiments to determine the optimal verbalizer, our method still outperforms it on all six tasks. These results demonstrate the efficacy of our method.



(a) Model for GPC $-W_F$ (b) Model for GPC $-W_R$ (c) Model for GPC $-W_R, W_F$

Fig. 3. Models in ablation study.

Table 3. Different settings for prompt update.

Model	Updating strategy
GPC	$P_{t+1} = \theta(W_F P_t^* + W_R P_t)$
GPC$-W_F$	$P_{t+1} = \theta(P_t^* + W_R P_t)$
GPC$-W_R$	$P_{t+1} = \theta(W_F P_t^* + P_t)$
GPC$-W_F, W_R$	$P_{t+1} = \theta(P_t^* + P_t)$

5 Ablation Study

Our GPC module consists of two parts, remembering unit and forgetting unit. The remembering unit receives previous prompt embeddings and selects which parts to retain. The remembering unit can be helpful for retaining certain information that requires long-term memory. The forgetting unit is responsible for determining which parts of the latest prompt embeddings are irrelevant, preventing the addition of unnecessary information.

We design the ablation experiments to explore the effectiveness of the above units. We conducted three groups of ablation experiments, where we added only the memory unit module, only the forgetting unit module, and no module at all. Through this, we aim to demonstrate the impact of memory and forgetting units on the experimental results, as well as prove the necessity and importance of these two modules.

Figure 3 shows the architectures of three ablation settings, which respectively remove the forgetting unit, the remembering unit and the both. Table 3 shows

Table 4. Ablation experiment results. GPC: Global Prompt Cell; GPC$-W_F$: removing the forgetting unit; GPC$-W_R$: removing the remembering unit; GPC-W_F, W_R: removing both the forgetting unit and remembering unit. **Bold**: the best performance.

		BoolQ	RTE	CB	COPA	WiC	WSC
BERT	GPC	**67.9**	**61.0**	**82.1**	**67.0**	**66.9**	**65.4**
	GPC$-W_F$	62.3	57.8	67.9	61.0	60.7	**65.4**
	GPC$-W_R$	62.6	57.8	55.4	63.0	57.5	64.4
	GPC$-W_F, W_R$	62.7	58.1	75.0	63.0	57.2	63.5
RoBERTa	GPC	**63.5**	**59.4**	**73.2**	**66.0**	**69.6**	**65.4**
	GPC$-W_F$	62.2	55.6	61.0	55.0	53.0	63.5
	GPC$-W_R$	62.2	52.7	64.3	55.0	50.9	63.5
	GPC$-W_F, W_R$	62.3	57.0	57.1	58.0	52.2	63.5

the formula of the three different updating strategies and the comparison with GPC.

The result in Table 4 shows that the contribution of two units varies from task to task, but the combination of the two can reach optimal results for all cases.

In addition to the findings presented in Table 4, further analysis of our ablation experiments revealed that the impact of the remembering and forgetting units on task performance was dependent on the specific task. Specifically, for some tasks, the addition of the memory unit module resulted in a greater improvement in performance compared to the addition of the forgetting unit module, while for other tasks the opposite was true.

The combination of both modules consistently led to optimal results across all tasks. This suggests that while the individual contributions of the remembering and forgetting units may vary depending on the specific task, the integration of both modules is crucial for achieving optimal performance across a range of tasks.

6 Conclusion

To conclude, our study has introduced the Global Prompt Cell (GPC) as a novel approach to enhance continuous prompt tuning. By selectively remembering and forgetting prompt embeddings, GPC enables more effective prompt updating, ultimately leading to improved model performance.

Through experiments conducted on the SuperGLUE benchmark, we have shown that our approach can substantially enhance results using prompt tuning. This highlights the potential of GPC to significantly improve the performance of language models on a range of natural language understanding tasks.

Finally, we emphasize that GPC can serve as a portable plug-in module for prompt tuning paradigms, allowing for easy integration with existing models and architectures. We believe that our approach has promising implications for the development of more effective and efficient language models, and we look forward to further exploration and refinement of this method in future research.

References

1. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding, pp. 4171–4186 (2019)
3. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3816–3830. Association for Computational Linguistics, Online (2021)
4. Gu, Y., Han, X., Liu, Z., Huang, M.: PPT: pre-trained prompt tuning for few-shot learning. arXiv preprint arXiv:2109.04332 (2021)
5. Hambardzumyan, K., Khachatrian, H., May, J.: WARP: word-level adversarial ReProgramming. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4921–4933. Association for Computational Linguistics, Online (2021)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **6**(02), 107–116 (1998)
8. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3045–3059 (2021)
9. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
10. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602 (2021)
11. Liu, X., et al.: GPT understands, too. arXiv preprint arXiv:2103.10385 (2021)
12. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
13. Qin, G., Eisner, J.: Learning how to ask: querying LMs with mixtures of soft prompts. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5203–5212. Online (2021)
14. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: a survey. SCI. CHINA Technol. Sci. **63**(10), 1872–1897 (2020)
15. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)

16. Schick, T., Schütze, H.: Exploiting cloze-questions for few-shot text classification and natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Association for Computational Linguistics, Online (2021)
17. Schick, T., Schütze, H.: It's not just size that matters: Small language models are also few-shot learners. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2339–2352. Association for Computational Linguistics, Online (2021)
18. Schmidhuber, J.: Deep learning in neural networks: an overview. CoRR abs/1404.7828 (2014)
19. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. D: Nonlinear Phenom. **404**, 132306 (2020)
20. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: AutoPrompt: eliciting Knowledge from Language Models with Automatically Generated Prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4222–4235. Association for Computational Linguistics, Online (2020)
21. Su, Y., et al.: On transferability of prompt tuning for natural language processing (2021)
22. Su, Y., et al.: On transferability of prompt tuning for natural language processing. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.naacl-main.290
23. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 194–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
24. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
25. Wang, A., et al.: SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019)
26. Wu, D., Wang, Y., Xia, S.T., Bailey, J., Ma, X.: Skip connections matter: on the transferability of adversarial examples generated with ResNets (2020)
27. Zhong, Z., Friedman, D., Chen, D.: Factual probing is [MASK]: learning vs. learning to recall. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5017–5033. Association for Computational Linguistics, Online (2021)

What Events Do Pre-trained Language Models Learn from Text? Probing Event-Based Commonsense Knowledge by Confidence Sorting

Jiachun Li[1,2], Chenhao Wang[1,2], Yubo Chen[1,2](✉), Kang Liu[1,2], and Jun Zhao[1,2]

[1] The Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{jiachun.li, chenhao.wang, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Abstract. Recently, there are a lot of works trying to probe knowledge in pre-trained language models (PLMs). Most probing works use data in knowledge bases to create a "fill-in-the-blank" task form and probe entity knowledge in auto-encoding PLMs (e.g. BERT). Though these works have got success, their methods can not be applied to some complicated knowledge like event-based commonsense knowledge and other PLMs such as auto-regressive models (like GPT). In this paper, we develop a new knowledge probe based on confidence sorting and detect event-based commonsense knowledge with it. To make the probe suitable for different types of PLMs, we integrate different knowledge scoring methods with a new method called probability difference log-likelihood score (PDL) among them. Finally, we conduct extensive experiments on several representative PLMs, explore their commonsense abilities and analyze the factors that influence their performances.

Keywords: Commonsense knowledge · Knowledge probing · Pre-trained language model

1 Introduction

Recently, large-scale pre-trained language models (PLMs) [6,14,15] have shown their strong abilities in many tasks [7,8,22]. Some models like GPT-3 [4] can even improve SOTA on some benchmarks in few-shot or zero-shot settings. Such performance indicates that these big models contain lots of knowledge after pre-training on the enormous corpus. Therefore, to explore what knowledge is in the model, many works began to use the "fill-in-the-blank" task to probe the entity knowledge in auto-encoding PLMs like BERT [1,3,10,13]. As an example, given the knowledge *(Dante, bornIn, Florence)*, these works mask the entity token

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 669–681, 2023.
https://doi.org/10.1007/978-3-031-44693-1_52

"Florence", put the converted sentence *"Dante was born in* [MASK]*"* into the model, and let the model predict the masked word. If the model successfully predicts the token *"Florence"*, they consider that the model has mastered this piece of knowledge.

Although these probing works have achieved great success, there are still some problems to be solved. On one hand, these works focus on entity knowledge, the entities of which mainly consist of words and simple phrases, such as *(Plane, IsCapbleOf, Fly)*. Nevertheless, concerning other knowledge that has complicated entities, the related research appears to be lacking. For example, event-based commonsense knowledge such as *(PersonX calls the police, xIntent, PersonX wants to report a crime)* in ATOMIC [17]. On the other hand, they mainly probe PLMs like BERT [6], other types of PLMs like GPT lack corresponding probing works. Hence, we focuses on these event-based knowledge and three different types of PLMs.

In this paper, we try to explore: (1) What level of event-based commonsense knowledge is contained in PLMs, (2) Whether there are differences in this knowledge across different types of PLMs, (3) What factors affect the level of this knowledge in PLMs. However, we find the traditional cloze task in probing works is not suitable enough for the event-based commonsense case. There are two main reasons: First, the concept in these knowledge is usually expressed by multiple tokens, so it is hard to create just one blank to fill in. e.g., given the knowledge *(PersonX calls the police, xWant, to report a crime)*, the tail entity *"report a crime"* can not be masked with one blank. Second, the concept in commonsense knowledge has diverse expressions, so there are multiple correct answers. For the former question, if we create the sentence *"PersonX calls the police, PersonX wants to* [MASK]*"*, the answer *"get some help"* and *"catch criminals"* are both suitable to fill in. Hence, a new probe method should be designed to detect these commonsense knowledge in PLMs.

To handle the above problems, firstly, we propose a knowledge probe for PLMs based on confidence sorting. Our probe takes the form of distinguishing positive and negative examples based on the confidence score of knowledge, instead of predicting the probability of one token. The overall framework of our work is shown in Fig. 1. In this process, a new method called probability difference log-likelihood score (PDL) is designed to reveal the commonsense abilities of auto-regressive models better compared to the previous methods. After scoring the knowledge with PLMs, the labels and scores are put into two different evaluation tasks, in order to estimate the commonsense abilities of the model. At last, we detect the event-based commonsense knowledge in several representative PLMs and explore the influence of different factors on the model's commonsense ability, including scoring methods, models' architecture and models' parameter size. Some meaningful conclusions can be derived from these experiments.

We summarize the contributions of the paper as follows:

(1) Since multi-word and multi-expression nature of commonsense knowledge brings many difficulties to the traditional cloze task, we design a new

confidence sorting probe to solve this problem, which successfully probes event-based commonsense knowledge in PLMs.

(2) While the former probing work is limited to BERT-like models, our probe can be applied not only to these auto-encoding models, but also to auto-regressive models (like GPT-2) and seq-to-seq models (like BART). Especially, we propose a knowledge scoring method called PDL in the probe to make better use of the knowledge in GPT-like models.

(3) We conduct comprehensive experiments with our probe, getting some meaningful conclusions around the PLM's commonsense knowledge ability. For example, PLMs' performance on different relation type, the effect of scoring methods on models' performance, the mechanism of model-related factors on models' commonsense knowledge level. More details will be discussed in the experiment section.

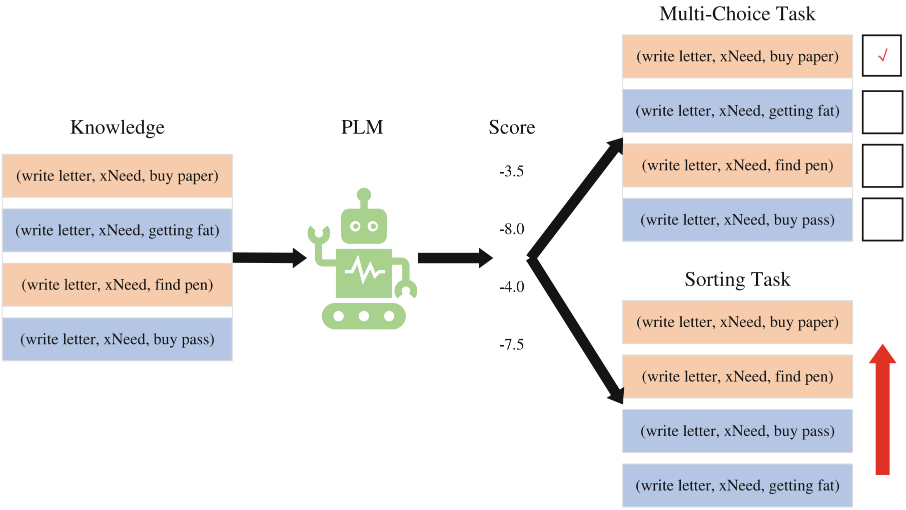

Knowledge
(write letter, xNeed, buy paper)
(write letter, xNeed, getting fat)
(write letter, xNeed, find pen)
(write letter, xNeed, buy pass)
PLM
Score
-3.5
-8.0
-4.0
-7.5
Multi-Choice Task
(write letter, xNeed, buy paper)
√
(write letter, xNeed, getting fat)
(write letter, xNeed, find pen)
(write letter, xNeed, buy pass)
Sorting Task
(write letter, xNeed, buy paper)
(write letter, xNeed, find pen)
(write letter, xNeed, buy pass)
(write letter, xNeed, getting fat)

Fig. 1. The process of the tasks in our probe, where the orange represents positive samples and the blue represents negative samples. Compared to the traditional multi-choice task, our sorting task can filter out multiple correct commonsense knowledge. (Color figure online)

2 Related Work

2.1 Probing Knowledge from PLMs

Since PLM became a new trend in NLP [19], knowledge probing work around it also get prevalent recently. In 2019, the LAMA probe [13] was the first to query the PLM for factual knowledge. By using a 'fill-in-the-blank' cloze statement, they trained BERT and ELMo to fill the true token into the mask blank and evaluate their knowledge level by the accuracy of prediction. This work brought a standard task form for knowledge probing tasks so there was much follow-up research based on it. For example, proposing a process to automatically optimize prompts in the LAMA [10], inducing relational knowledge from BERT by classifying the complete sentence after filling the blank [3], and using cloze tasks to probe BERT for noun properties knowledge [1].

While these works probed different kinds of knowledge in BERT, the work around commonsense knowledge seems to be lacking because of the multiple tokens and diverse expressions in their commonsense concepts. We successfully probe commonsense knowledge in PLMs by creating a probe based on knowledge confidence sorting. Moreover, most probing works focused on BERT-like models, covering only one type of PLMs. We extend the probing work to more types of models and probe event-based commonsense knowledge from them.

2.2 Automatic Acquisition of Commonsense Knowledge

Commonsense knowledge also got much attention these years because of its significance for reasoning tasks in NLP. Many works gathered a lot of commonsense knowledge to make research on it. In the beginning, these works acquired commonsense knowledge through artificial methods like crowdsourcing. For example, ConceptNet [18], ATOMIC [17], ATOMIC$^{20}_{20}$ [9] and so on. However, these knowledge acquisition processes rely on humans, resulting in a high cost and low efficiency. Therefore, to make up for these deficiencies, several kinds of tasks have been designed for the automatic acquisition of commonsense knowledge. For example, commonsense knowledge base completion tasks [2,12], commonsense knowledge generation tasks [5,20,23] and commonsense knowledge evaluation tasks [24,25]. Though these works take different research objectives and forms, they all realize the automatic acquisition of commonsense knowledge by mining the knowledge in language models, especially large-scale PLMs.

Our work is the continuation and development of the above works. We propose a new commonsense evaluation task based on knowledge sorting. Besides, we comprehensively analyze various factors that affect the commonsense level of the PLM, complementing the lack of related work.

3 The Commonsense Knowledge Confidence Sorting Probe

We introduce our probe to detect event-based commonsense knowledge in diverse PLMs. First, we extract data from ATOMIC$^{20}_{20}$. Then, we transform the data into an appropriate form by generating negative samples from these facts and converting all the knowledge triples into sentences. After getting confidence scores with scoring methods, we can estimate the level of commonsense knowledge in PLMs through evaluation tasks and metrics. The design of our probe is discussed in detail below.

3.1 Knowledge Source

ATOMIC$^{20}_{20}$ was used as the commonsense knowledge resource for our probing work, for the reason that it's the one of the largest commonsense knowledge base at present. The knowledge base classified commonsense knowledge into three categorical types: physical-entity knowledge, event-centered knowledge, and social-interaction knowledge. As many related works focused on the entity knowledge [1,3], our experiments are conducted on the latter two, which cover 15 event-based relations (we remove the *isFilledBy* relation because of its bad format). To give a explicit definition of the knowledge we study, each piece of it consists of two entities describing events and a phrase representing the relation between the two events (e.g. *(PersonX turns off the light, isBefore, PersonX sleeps)*). The whole knowledge triple denotes some commonsense in our daily life, so we call it event-based commonsense knowledge.

3.2 Negative Sampling

As for distinguishing positive and negative examples, we need to generate negative examples. Given a positive example $\tau = (h, r, t)$, the following three strategies are used to construct two negative examples for each τ. This part mainly references Li's Work [12].

Random Sampling. We randomly select two other tuples $\tau' = (h', r', t')$ and $\tau'' = (h'', r'', t'')$ from the dataset. Then, we replace h with h' and replace t with t''. r won't be replaced because we must fix the numbers of knowledge in each relation.

Relation Sampling. The commonsense knowledge in the same relation is usually similar in syntax and grammar. In that case, we restrict the scope of the former replacement to the same relation, to reduce format errors in negative examples.

Max Sampling. We create four negative examples through two previous strategies. After that, we give the log-likelihood score for these triples with the GPT-2 model [14]. We choose two examples that have the two highest scores to pursue more high-quality negative examples.

3.3 Converting Triples to Sentences

Like the former work [5], we use the template filling form to realize the conversion. For each relation in our dataset, we manually design several templates, then fill the head and tail of the knowledge into them. For example, one template for the relation *xNeed* is: *"PersonX needed* [Tail] *before* [Head]". For the tuple $\tau =$ *(write letter, xNeed, buy paper)* in Fig. 1, we generate the sentence: *"PersonX need buy paper before write letter"*. We fill the triple into all of the corresponding templates, form a set of candidate sentences, and choose the sentence with the highest log-likelihood score as the output following Davison's work [5].

3.4 Evaluation Tasks

Since we have successfully got scores and labels for each knowledge example through past steps, then we begin to design evaluation tasks and metrics to estimate the commonsense knowledge level in models.

Multi-choice Task. The multi-choice task is widely taken in previous literature [21,24,25], here we also use it in our probe as a comparison to sorting task. For a positive example and two negative examples, we argue that models predict the highest-scoring example as the positive one. Thus we can get the accuracy of the model's prediction, assessing the commonsense abilities of models.

Sorting Task. Since concepts in commonsense knowledge have diverse expressions, it seems impossible there is only one true positive example in practice. So we should study the ability of the model to distinguish between multiple positive and negative examples. That's why we design the sorting task. In this task, we put all the labels and confidence scores of examples in the same relation together and sort examples by scores. We take AUC as our metric, which can calculate the probability that a positive example comes before a negative one by dynamically adjusting the threshold. We assume that the deeper commonsense ability of models can be evaluated through this harder task and we will give a proof of its effectiveness in Sect. 5.1.

4 Scoring Knowledge with PLMs

For our probe to work successfully, the confidence score of commonsense knowledge is of great necessity. Thus, we'd like to introduce the knowledge scoring methods for PLMs in this section. The PLM based on the Transformer has three main architecture types: auto-encoding models, auto-regressive models, and seq-to-seq models. Our research implement scoring methods for each type.

4.1 Auto-Encoding Models

Since most of this type of models are masked language models (MLMs), we use the masking mechanism to calculate the confidence score of the knowledge. Two scores are as follows:

Pseudo-Log-Likelihood Score (PLL). We calculate the probability of the knowledge by summing the conditional log probabilities $logP(t_i|S_{\backslash t_i})$, which are induced by replacing token t_i in knowledge sentence S with $[MASK]$ in the MLM [16]. The score is:

$$PLL(S) = \sum_{i=1}^{|S|} logP(t_i|S_{\backslash t_i})$$

Point-Wise Mutual Information (PMI). Given a commonsense triple $\tau = (h, r, t)$, considering the correlation between two entities [5], defined as:

$$PMI(t, h|r) = logP(t|h, r) - logP(t|r)$$

Like the first score, we leverage the masking mechanism in the MLM to calculate the conditional probability P. Besides, we also compute the score $PMI(h, t|r)$ by the symmetrical formula and average two scores as the final score.

4.2 Auto-Regressive Models

These models use the auto-regressive mechanism to compute the probability P. Here are two scores:

Perplexity (PPL). PPL is a common metric for evaluating language models. Let S denotes a tokenized sequence $(t_0, t_1, \cdots t_n)$, where t_0 is the start token, the formula as below:

$$PPL(S) = exp\left\{-\frac{1}{n}\sum_{i=1}^{n} logP(t_i|t_{<i})\right\}$$

Note that the score is negatively correlated with the probability of sentence, we take the inverse of it when using.

Probability Difference Log-Likelihood Score (PDL). In the traditional PPL method, the probabilities of entities themselves can largely make a difference to the confidence score of the whole knowledge. We find the triple that contains more function words in the entities tends to get a higher score. For example, compared to the wrong knowledge *"Death cause they don't have a car to drive in"*, the true knowledge *"Death causes decomposition"* has a lower score when using PPL. To reduce that noise, we put forward a score called PDL, which subtracts the probability of the entity itself from the calculation. Given a commonsense triple $\tau = (h, r, t)$, the calculation is as below:

$$PDL(t|h, r) = logP(t|h, r) - logP(t) \quad PDL(h|r, t) = logP(h|r, t) - logP(h)$$

To capture bi-directional association information of the head and tail, we create bi-directional templates for generating sentences. The final PDL is obtained by averaging the above two.

4.3 Seq-to-Seq Models

Most of the models here take a sequence-to-sequence architecture with the masking mechanism on the encoder side and auto-regressive mechanism on the other side, such as BART [11], T5 [15] and so on. Thus, former scores can be used by these models.

5 Experiments

As an application of our probe, we choose several representative models, detect the event-based commonsense knowledge involved and identify factors that affect their commonsense abilities. All of the experiments in this section are performed without the fine-tuning stage.

5.1 The Effectiveness of the Sorting Task

We make the comparison of two methods on the GPT2-xl model, whose results are shown in Fig. 2. As we can see, there is consistency in the trend of the two broken lines. The relation types that the model performs well in the multi-choice task also perform equally well in the sorting task, and vice versa. Since the multi-choice method has been widely adopted in related works, we can conclude that our sorting task can truly detect the commonsense level in PLMs, too. To make our results more reliable, we also repeat the same experiments on other five models (e.g. BERT, RoBerta, BART) and get the same results.

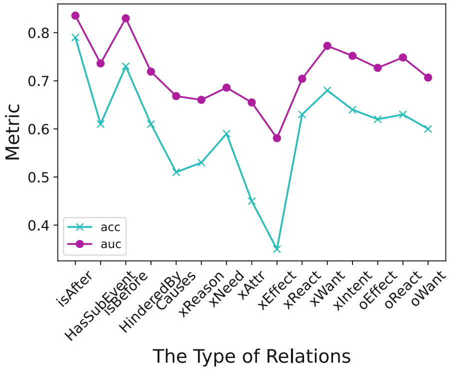


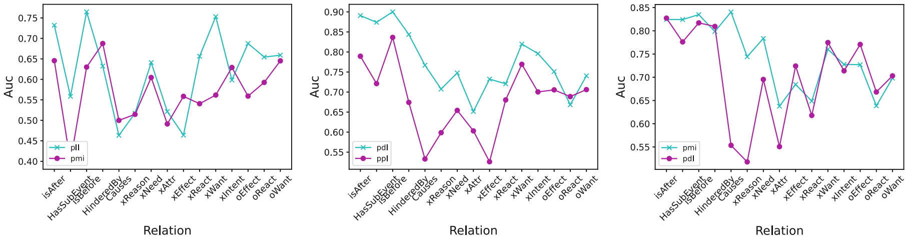
Fig. 2. Two evaluation tasks' metrics on 15 relation types in our dataset. The blue line represents metrics of the multi-choice task and the red line represents the sorting task. (Color figure online)

5.2 The Difference Between Relation Types

From Fig. 2, we can conclude that the performance on different relations has a big difference in both tasks. Thus, we want to explore models' advantageous and disadvantaged relation types in commonsense knowledge. For each model, we change the scoring method, negative sampling strategy, and model's version, calculating the average AUC in the sorting task. The result is demonstrated in Table 1. It is clear that different PLMs have different advantageous and disadvantaged relation types. For BERT, relations like *xWant* and *IsAfter* get better performance, while *Causes* and *xAttr* get worse performance. For GPT-2, relations like *IsAfter* and *IsBefore* get better performance, while *xEffect* and *xAttr* get worse performance. For BART, relations like *HasSubEvent* and *IsBefore* get better performance, while *xAttr* and *xReact* get worse performance. Besides, there are some relation types that all models are good or bad at learning. For example, timing relations like *IsAfter* or *IsBefore* gets good performance, causation relations like *Causes* gets bad performance. We infer that it is because the timing relation usually has explicit linking words in the pre-training text (e.g. *"then"*, *"before"*), while the relation type like *xAttr* does not have. For models, this character makes the former types of knowledge easier to learn.

Table 1. The average AUC for 15 relations on the sorting task.

Models	IsAfter	HasSubEvent	IsBefore	HinderedBy	Causes	xReason	xNeed	xAttr
BERT	0.636	0.521	0.629	0.607	0.444	0.537	0.547	0.527
GPT-2	0.743	0.709	0.743	0.684	0.623	0.640	0.613	0.611
BART	0.693	0.723	0.713	0.711	0.637	0.617	0.677	0.594
	xEffect	xReact	xWant	xIntent	oEffect	oReact	oWant	
BERT	0.587	0.570	0.645	0.574	0.614	0.579	0.629	
GPT-2	0.602	0.632	0.682	0.689	0.630	0.647	0.642	
BART	0.646	0.601	0.681	0.663	0.657	0.616	0.642	



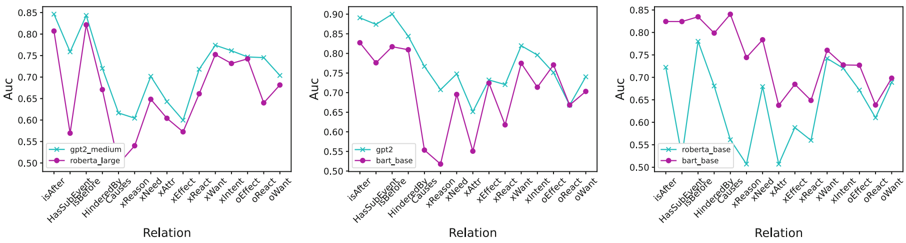
(a) BERT-base (b) GPT2 (c) BART-base

Fig. 3. Results of the influence of scoring methods on the sorting task. The comparison experiment is conducted on BERT-base, GPT2, and BART-base.

5.3 The Comparison of Scoring Methods

As various methods have been taken in the model scoring stage, we would like to compare the difference between them. Here we select three models with different architectures: BERT, GPT-2, and BART, comparing different scores' performance on the sorting task, whose results are shown in Fig. 3. According to the figure, our probe can compare different score to find the method that reflects the model's commonsense level better. Therefore, according to Fig. 3, we find that our new method PDL can exploit the commonsense knowledge in GPT-like models more fully, compared to the traditional PPL method.

We also make a verification in the ComVE's subtask B [21] with the GPT2-xl model to certify the applicability of the PDL. Compared to PPL's **0.588** accuracy, our PDL get **0.604** accuracy, which indicates that this method is capable of better evaluating the commonsense level of PLMs in other commonsense-related tasks.

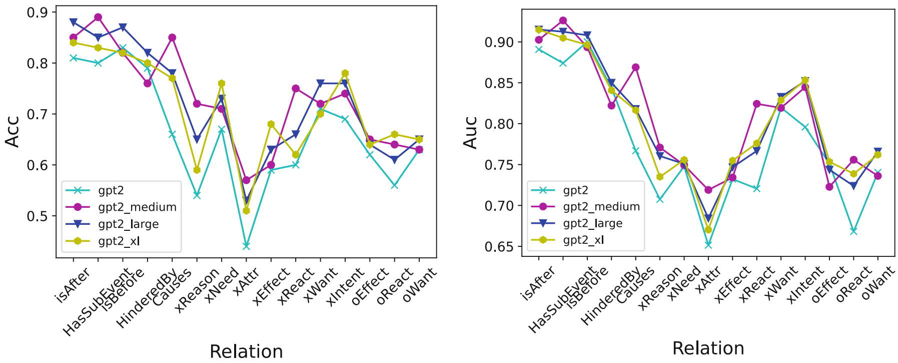


(a) GPT-2 and RoBERTa (b) GPT-2 and BART (c) RoBERTa and BART

Fig. 4. Results of the influence of models' architecture on the sorting task.

5.4 The Influence of Model Attributes

Since we introduce diverse PLMs in this work, the influence of the model's attributes should also be studied. Here we select two main factors to study: the model's architecture and the size of the model's parameters.



(a) Multi-choice Task (b) Sorting Task

Fig. 5. Results of the influence of models' parameter size on two evaluation tasks.

Since we use models with different architectures, it is necessary to explore the effect of the architecture on commonsense abilities. For reducing the interference from other facts, we use identical scoring methods and versions with similar parameter sizes. This paper studies three pairs of models here: GPT2-medium (354M parameters) and RoBERTa-large (355M parameters), GPT2 (124M parameters) and BART-base (139M parameters), RoBERTa-base (125M parameters) and BART-base (139M parameters). The results of scoring tasks are shown in Fig. 4. It can be inferred that with similar numbers of the parameter, auto-regressive architecture can learn commonsense knowledge better, while auto-encoding architecture performs worse. This may be because that our task form of scoring sentences is more suitable for the pre-training task of the autoregressive model, but does not match the pre-training task of the latter.

Nowadays, the size of parameters in PLMs is constantly increasing, we want to explore whether the larger parameter size brings more commonsense knowledge. We use different versions of GPT-2 and compare their performances on two tasks, whose result is shown in Fig. 5. From the figure, it can be learned that as the number of parameters rises, the commonsense ability of the model first rises and remains almost constant or even declines in some relations. We infer that there is a saturation state in the commonsense ability of the model. When the parameter numbers are large enough to get that state, the model cannot further improve the capability only by pre-training. Moreover, bigger models are more sensitive to the false negative examples in our experiments, causing some declines. Therefore, the size of models' parameters is not positively correlated with the commonsense level, and this factor cannot determine the commonsense level alone.

6 Conclusion

In this paper, we propose a commonsense knowledge confidence sorting probe, effectively detecting event-based commonsense knowledge abilities of PLMs from different architectures. Besides, a new knowledge scoring method called PDL is designed in our probe, which calculates the confidence score better compared to the previous method. Finally, our work finds out factors that affect the model's commonsense abilities and analyzes the mechanism of their influence.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106400), the National Natural Science Foundation of China (No. 61976211, 62176257). This work is also supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA27020100), the Youth Innovation Promotion Association CAS, and Yunnan Provincial Major Science and Technology Special Plan Projects (No. 202202AD080004).

References

1. Apidianaki, M., Soler, A.G.: ALL dolphins are intelligent and SOME are friendly: probing BERT for nouns' semantic properties and their prototypicality. CoRR abs/2110.06376 (2021)
2. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: commonsense transformers for automatic knowledge graph construction. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) ACL 2019, Florence, Italy, 28 July–2 August 2019, Volume 1: Long Papers, pp. 4762–4779 (2019)
3. Bouraoui, Z., Camacho-Collados, J., Schockaert, S.: Inducing relational knowledge from BERT. In: AAAI 2020, IAAI 2020, EAAI 2020, New York, NY, USA, 7–12 February 2020, pp. 7456–7463 (2020)

4. Brown, T.B., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 6–12 December 2020, Virtual (2020)
5. Davison, J., Feldman, J., Rush, A.M.: Commonsense knowledge mining from pretrained models. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 1173–1178 (2019)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
7. Du, H., Le, Z., Wang, H., Chen, Y., Yu, J.: COKG-QA: multi-hop question answering over COVID-19 knowledge graphs. Data Intell. **4**(3), 471–492 (2022)
8. Gong, Y., Mao, L., Li, C.: Few-shot learning for named entity recognition based on BERT and two-level model fusion. Data Intell. **3**(4), 568–577 (2021)
9. Hwang, J.D., et al.: (comet-) atomic 2020: on symbolic and neural commonsense knowledge graphs. In: AAAI 2021, IAAI 2021, EAAI 2021, Virtual Event, 2–9 February 2021, pp. 6384–6392 (2021)
10. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know. Trans. Assoc. Comput. Linguist. **8**, 423–438 (2020)
11. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) ACL 2020, Online, 5–10 July 2020, pp. 7871–7880 (2020)
12. Li, X., Taheri, A., Tu, L., Gimpel, K.: Commonsense knowledge base completion. In: ACL 2016, 7–12 August 2016, Berlin, Germany, Volume 1: Long Papers, pp. 1445–1455 (2016)
13. Petroni, F., et al.: Language models as knowledge bases? In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 2463–2473 (2019)
14. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
16. Salazar, J., Liang, D., Nguyen, T.Q., Kirchhoff, K.: Masked language model scoring. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) ACL 2020, Online, 5–10 July 2020, pp. 2699–2712 (2020)
17. Sap, M., et al.: ATOMIC: an atlas of machine commonsense for if-then reasoning. In: AAAI 2019, IAAI 2019, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 3027–3035 (2019)
18. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: an open multilingual graph of general knowledge. In: Singh, S., Markovitch, S. (eds.) AAAI 2017, 4–9 February 2017, San Francisco, California, USA, pp. 4444–4451 (2017)
19. Sun, T.X., Liu, X.Y., Qiu, X.P., Huang, X.J.: Paradigm shift in natural language processing. Mach. Intell. Res. **19**(3), 169–183 (2022)
20. Wang, C., Li, J., Chen, Y., Liu, K., Zhao, J.: CN-automic: distilling Chinese commonsense knowledge from pretrained language models. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022, pp. 9253–9265. Association for Computational Linguistics (2022)

21. Wang, C., Liang, S., Jin, Y., Wang, Y., Zhu, X., Zhang, Y.: SemEval-2020 task 4: commonsense validation and explanation. In: Herbelot, A., Zhu, X., Palmer, A., Schneider, N., May, J., Shutova, E. (eds.) COLING 2020, Barcelona (online), 12–13 December 2020, pp. 307–321 (2020)
22. Wang, X., et al.: Large-scale multi-modal pre-trained models: a comprehensive survey. Mach. Intell. Res. **20**(4), 447–482 (2023)
23. West, P., et al.: Symbolic knowledge distillation: from general language models to commonsense models. CoRR abs/2110.07178 (2021)
24. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: HellaSwag: can a machine really finish your sentence? In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) ACL 2019, Florence, Italy, 28 July–2 August 2019, Volume 1: Long Papers, pp. 4791–4800. Association for Computational Linguistics (2019)
25. Zhou, X., Zhang, Y., Cui, L., Huang, D.: Evaluating commonsense in pre-trained language models. In: AAAI2020, IAAI 2020, EAAI 2020, New York, NY, USA, 7–12 February 2020, pp. 9733–9740 (2020)

Revisit Input Perturbation Problems for LLMs: A Unified Robustness Evaluation Framework for Noisy Slot Filling Task

Guanting Dong[1], Jinxu Zhao[1], Tingfeng Hui[1], Daichi Guo[1], Wenlong Wang[1], Boqi Feng[1], Yueyan Qiu[1], Zhuoma Gongque[1], Keqing He[2], Zechen Wang[1], and Weiran Xu[1](✉)

[1] Beijing University of Posts and Telecommunications, Beijing, China
{dongguanting,zhaojinxu,huitingfeng,guodaichi,zechen_wang, fbq,yuanqiu,wanwenlong,xuweiran}@bupt.edu.cn
[2] Meituan Group, Beijing, China
hekeqing@meituan.com

Abstract. With the increasing capabilities of large language models (LLMs), these high-performance models have achieved state-of-the-art results on a wide range of natural language processing (NLP) tasks. However, the models' performance on commonly-used benchmark datasets often fails to accurately reflect their reliability and robustness when applied to real-world noisy data. To address these challenges, we propose a unified robustness evaluation framework based on the slot-filling task to systematically evaluate the dialogue understanding capability of LLMs in diverse input perturbation scenarios. Specifically, we construct a input perturbation evaluation dataset, Noise-LLM, which contains five types of single perturbation and four types of mixed perturbation data. Furthermore, we utilize a multi-level data augmentation method (character, word, and sentence levels) to construct a candidate data pool, and carefully design two ways of automatic task demonstration construction strategies (instance-level and entity-level) with various prompt templates. Our aim is to assess how well various robustness methods of LLMs perform in real-world noisy scenarios. The experiments have demonstrated that the current open-source LLMs generally achieve limited perturbation robustness performance. Based on these experimental observations, we make some forward-looking suggestions to fuel the research in this direction (The code is available at https://github.com/ZhaoJin-xu/A-Unified-Robustness-Evaluation-Framework-for-Noisy-Slot-Filling-Task).

Keywords: Large language models · Robustness evaluation · Slot filling · Input perturbation

G. Dong and J. Zhao—The first two authors contribute equally.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 682–694, 2023.
https://doi.org/10.1007/978-3-031-44693-1_53

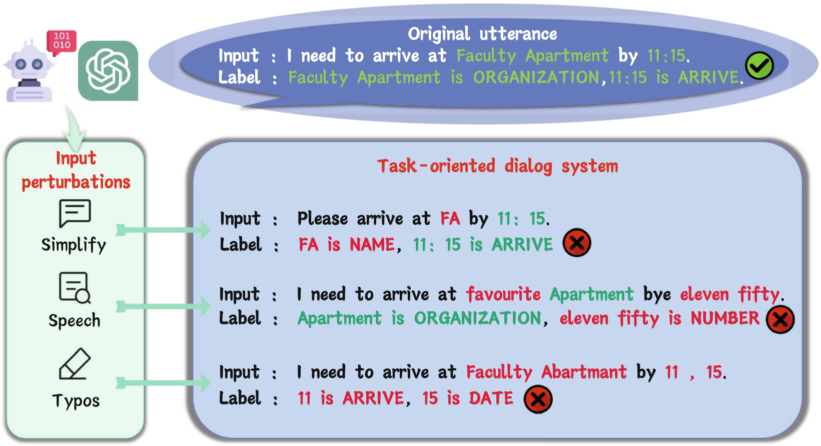

101 010
Original utterance
Input : I need to arrive at Faculty Apartment by 11:15.
Label : Faculty Apartment is ORGANIZATION,11:15 is ARRIVE.
Input perturbations
Simplify
Speech
Typos
Task-oriented dialog system
Input : Please arrive at FA by 11: 15.
Label : FA is NAME, 11: 15 is ARRIVE
Input : I need to arrive at favourite Apartment bye eleven fifty.
Label : Apartment is ORGANIZATION, eleven fifty is NUMBER
Input : I need to arrive at Facullty Abartment by 11 , 15.
Label : 11 is ARRIVE, 15 is DATE

Fig. 1. The impact of various types of input perturbations on the slot filling system in real-word scenarios

1 Introduction

The slot filling (SF) task in the goal-oriented dialog system aims to identify task-related slot types in certain domains for understanding user utterances. Recently, Large-scale language models (LLMs) [2,16,23] have shown an impressive ability for in-context learning with only a few task-specific examples as demonstrations. Under the framework of in-context learning, LLMs have achieved promising results in a variety of NLP tasks, including machine translation (MT) [20], question answering (QA) [9] and named entity extraction (NEE) [2]. However, despite its powerful abilities, the high performance of these models depend heavily on the distribution consistency between training data and test data. As the data distribution of dialogues in real scenarios is unknown [25], there are still many challenges in applying these methods to real dialogue scenarios.

In real dialogue systems, due to the diverse language expression and input errors of humans, models often need to deal with a variety of input perturbations. As shown in Fig. 1, due to different expression habits, users may not interact with the dialogue system following the standard input format, but may simplify their queries to express the same intent. What's more, errors in the upstream input system may also introduce disturbances to the downstream model (e.g. typos from keyboard input, speech errors from ASR systems). LLMs are usually pre-trained and fine-tuned on perturbation-free datasets, resulting in poorer performance than supervised small language models with robust settings on noisy slot filling tasks. However, the accuracy of slot filling tasks directly reflects the model's understanding of user queries, which would impact the performance of the model in other downstream tasks. Therefore, exploring the robustness and generalization of LLMs under various input perturbations is crucial for the application of task-oriented dialogue systems in real scenarios.

To address the above challenges, in this paper, we aim to investigate how well various robustness methods of large language models perform in real-world noisy scenarios, and then provide empirical guidance for the research of robust LLMs. Based on the slot filling task, we propose a unified robustness evaluation framework to evaluate the dialogue understanding capability of large models in diverse input perturbation scenarios. Firstly, we construct a input perturbation evaluation dataset, Noise-LLM, to investigate the robustness of large language models in two different input perturbation settings. 1) **Single perturbation setting:** Based on the DST dataset Raddle [18] which contains various types of real-world noisy texts, we transformed them into slot filling data through manual annotation which consists of 5 types of single perturbation at the character, word, and sentence levels. 2) **Mixed perturbation setting:** We utilized the widely used Slot Filling dataset SNIPS [3] and data augmentation tools [6] to construct 4 types of mixed perturbation that fit real-world dialog scenarios. Furthermore, we utilize a multi-level data augmentation method (character, word, and sentence levels) to construct a candidate data pool, and carefully designed two ways of automatic task demonstration construction strategies (Instance-level and Entity-level) with various prompt templates. Our goal is to provide empirical guidance for research on robust LLMs based on our evaluation framework. Based on this framework, we conduct extensive experiments on Noise-LLM and analyze the results from the perspectives of input perturbation types and different demonstrations, respectively.

Our main contributions are concluded as follows:

(1) To our best knowledge, we are the first to comprehensively investigate the effects of input perturbations at different levels on LLMs, and construct an input perturbation evaluation dataset, Noise-LLM, which includes single perturbation and mixed perturbation settings that fit real-world scenarios.
(2) We propose a unified robustness evaluation framework based on the SF task, which includes a multi-level data augmentation method, diverse prompt templates, and various automatic task demonstration construction strategies.
(3) Experiments result demonstrate that the current open-source LLMs generally have limited ability to counter input perturbations. The extensive analysis also provides empirical guidance for further research.

2 Related Work

2.1 Input Perturbation Problem

Owing to the notable performance gaps between benchmarks and real-world scenarios, the robustness of NLP systems in the face of input perturbations has garnered significant attention recently. [10,14] conduct empirical evaluations of the robustness of various NLP systems against input perturbations using synthetic benchmarks that we generated. [4,7] focus on the robustness of the sequence labeling framework against optical input perturbations (e.g. misspellings, OOV).

[5] investigate the robustness of dialogue systems on ASR noise. [21] mainly focus on the ASR noise-robustness SLU models in dialogue systems. In this paper, we investigate the abilities of ChatGPT against input perturbation problems for slot filling tasks.

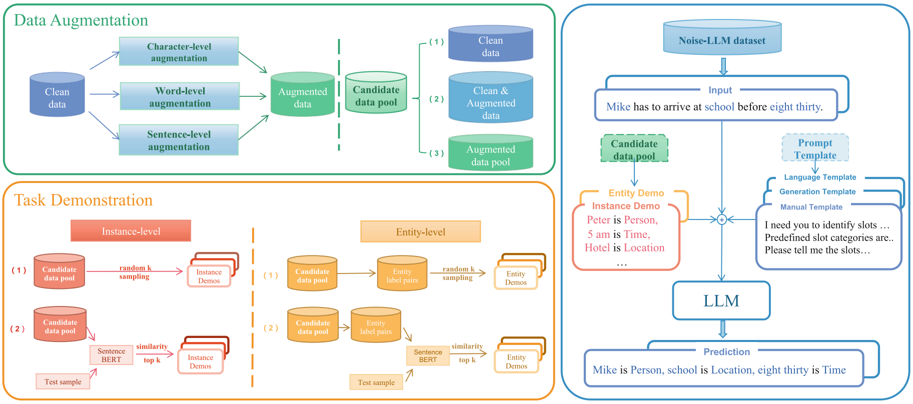


Fig. 2. The overall architecture of our unified robustness evaluation framework based on slot filling task

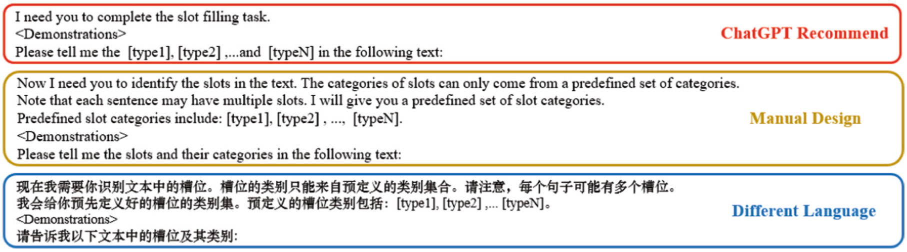
2.2 Large Language Models

The emergence of large-scale language models (LLMs) have brought revolutionary transformation to the field of natural language processing (NLP) [22]. LLMs, such as GPT-3 [2], LLaMA [23], ChatGPT, and GPT-4 [16], have demonstrated impressive abilities on various tasks and in generating fluent responses due to the large-scale training corpora, as well as the use of external techniques such as instruction tuning and reinforcement learning from human feedback (RLHF) [17]. LLMs based on generative framework reformulate the information extraction task [11], task-oriented dialog systems [19,26], even from multi modal perspective [12,27,28]. More recently, the NLP community has been exploring various application directions for LLMs. For instance, chain-of-thought prompting [24] enables LLMs to generate problem-solving processes step-by-step, significantly enhancing the model's reasoning ability. Some researchers have utilized the powerful interactive capabilities of LLMs to generate commands that invoke external tools for better handling of various downstream tasks [22]. Other researchers have proposed parameter-efficient fine-tuining (PEFT) methods to address the issue of excessive computational resource usage during fine-tuning of LLMs, which significantly reduces resource consumption without sacrificing performance [8]. In this paper, we aim to investigate the robustness of LLMs against input perturbations. To be specific, we explore several prompt and demonstration strategies and conduct a large number of experiments to advance our research.

3 Method

3.1 Problem Definition

Given an input utterance $X = \{x_1, x_2, ..., x_N\}$, where N represents the length of X, we adopt a triple $y_i = \{l, r, t\} \in Y$ to represent the $i - th$ entity that appears in X, where Y represents all the entity triplets in X, and l, r denote the entity boundaries, while t denotes the entity type. We use D_{clean}, $D_{augment}$, and D_{test} to denote the conventional labeled slot filling datasets, including the clean dataset, the data-augmented dataset, and the test set, respectively, where $D_{test} = \{X_1 : Y_1, X_2 : Y_2, ...X_N : Y_N\}$. We define $P_i = X_i : Y_i$, and then $D_{test} = P_1, P_2, ..., P_N$, and we formulate the spoken language perturbation process in the real scenario as $\widetilde{P} = \Gamma(P)$, such that $\widetilde{X} \neq X$ but $\widetilde{Y}$ may or may not be identical to Y. The form of the perturbation transformation Γ is flexible, which could be a specific type of perturbation (e.g., Typos) or mixed perturbation.Based on the above process, we can obtain a perturbed test set $\widetilde{D}_{test}$.



I need you to complete the slot filling task.
<Demonstrations>
Please tell me the [type1], [type2] ,...and [typeN] in the following text:

Now I need you to identify the slots in the text. The categories of slots can only come from a predefined set of categories.
Note that each sentence may have multiple slots. I will give you a predefined set of slot categories.
Predefined slot categories include: [type1], [type2] , ..., [typeN].
<Demonstrations>
Please tell me the slots and their categories in the following text:

现在我需要你识别文本中的槽位。槽位的类别只能来自预定义的类别集合。请注意，每个句子可能有多个槽位。
我会给你预先定义好的槽位的类别集。预定义的槽位类别包括：[type1], [type2] ,... [typeN]。
<Demonstrations>
请告诉我以下文本中的槽位及其类别:

Fig. 3. The different prompt templates for slot filling task.

3.2 Data Augmentation

As shown in the top left corner of Fig. 2, we utilize the data augmentation tool NLPAug [13] to transform the clean training set into an augmented dataset. The specific data augmentation methods employed in this study can be categorized into the following three types: **Character-level augmentation:** randomly add, delete and replace characters in one token with the probability p. **Word-level augmentation:** randomly delete and insert words and replace words with homophones in one sentence with probability p. **Sentence-level augmentation:** replace sentences with synonymous ones.

3.3 Task Demonstration

Task demonstration D is constructed by retrieving instance examples from a dataset. The role of constructing task demonstration is to select appropriate examples that can effectively demonstrate how the LLM should solve the task. We categorize the demonstration into two types: (1) Entity-oriented demonstration, and (2) Instance-oriented demonstration.

Entity-Oriented demonstration. Given a set of entity type labels $L = l_1, l_2, ..., l_N$, where N is the number of label L, we select one entity example e per label l from D_{train} and modify it into the form [e_i is l_i]. For each label l, we adopt two strategies for selecting entity e: (1) Randomly selecting entity e from the clean, augmented, or mixed data pool. (2) Retrieving the k most relevant entity examples e from the clean, augmented, or mixed data pool using SentenceBERT [20]. Specifically, the SentenceBERT algorithm generates independent CLS embeddings for the input x and e, and calculates the cosine similarity between them to rank the entity e.

Instance-Oriented Demonstration. Given an example S and its label Y, we modify all of its entities and their corresponding labels into the form of "e_i is l_i", and concatenate them with S to form S'. We also adopt two strategies to select example S: (1) Randomly selecting example S from the clean, augmented, or mixed data pool. (2) Retrieving the k most relevant examples S from the clean, augmented, or mixed data pool using SentenceBERT.

Prompt Template. ChatGPT is a large language model that relies on prompts to guide its output generation. The quality of the output can be influenced by the style of the prompts used. Since slot filling is essentially a sequence labeling task rather than a generation task, it is not entirely consistent with the paradigm of contextual learning. Therefore, we have designed three templates specifically for slot filling. To design these prompts, we initially sought inspiration from ChatGPT by requesting its advice. However, since the templates provided by ChatGPT were not specifically tailored to our task, we felt that this could potentially impact its performance. Hence, we also designed prompts based on the requirements of slot filling. Considering that ChatGPT is primarily pre-trained on English language data, its understanding of prompts in other languages may vary. Therefore, we also tested the English templates translated into Chinese using ChatGPT. The three prompts used in our experiments are shown in Fig. 3.

3.4 In-Context Inference

In-context learning (ICL) is popularized as a way of learning for large language models in the original GPT-3 paper [1]. When using in-context learning, we give the large language model a task description and a set of demonstration examples (input-label pairs). A test input is then appended to the end of the demonstration examples, and the LM is allowed to make predictions only based on the conditioned demonstration examples. In order to answer the question correctly, the model needs to understand the demonstration examples of ICL to determine the input distribution, output distribution, input-output mapping, and format. We concatenate the prompt, demonstration, and test input for in-context learning and use it as model input for inference.

4 Experiments

4.1 Datasets

Based on RADDLE [18] and SNIPS [3], we constructed the Noise-LLM dataset, which includes two settings: single perturbation and mixed perturbation. For the single perturbation setting, RADDLE is a crowdsourced diagnostic evaluation dataset covering a broad range of real-world noisy texts for dialog systems. We extract 50 utterances for clean data, and each type of noisy utterance (Typos, Speech, Simplify, Verbose, and Paraphrase) from RADDLE to construct the evaluation set. **Typos** are caused by non-standard abbreviations, while **Speech** arises from recognition and synthesis errors from ASR systems. **Simplification** refers to users using concise words to express their intentions, while **Verbose** refers to users using redundant words to express the same intention. **Paraphrase** is also common among users who use different words or restate the text based on their language habits. For mixed perturbation setting, based on SNIPS, we used textflint [6] to introduce character-level perturbation **Typos**, Word-level perturbation **Speech**, sentence-level perturbation **AppendIrr**, and mixed perturbation to the test set and construct a multi-perturbation evaluation set.

Table 1. The performance (F1 score) of the finetuned SOTA (NAT, PSSAT) and LLMs (Text-davinci-003, ChatGPT) with best entity-oriented and instance-oriented demonstrations on clean and noisy test sets.

Methods	Clean	Character	Word	Sentence			Overall
		Typos	Speech	Paraphrase	Simplification	Verbose	
NAT($\mathcal{L}_{aug}$)	96.01	67.47	85.23	87.73	87.32	85.41	87.21
NAT($\mathcal{L}_{stabil}$)	96.04	67.54	85.16	87.42	87.33	85.29	87.27
PSSAT	96.42	68.34	85.65	91.54	89.73	85.82	88.16
Text-davinci-003	43.09	34.26	39.34	38.42	40.12	37.18	38.54
ChatGPT	71.43	40.65	60.00	55.56	65.54	55.56	57.21
ChatGPT+Instance level	68.21 (−3.2)	65.04 (+24.3)	70.56 (+10.5)	58.82 (+2.2)	73.02 (+7.4)	61.77 (+6.2)	68.34 (+11.1)
ChatGPT+Entity level	74.07 (+2.6)	62.18 (+21.5)	55.39 (+4.6)	75.59 (+18.9)	70.96 (+5.4)	71.75 (+16.1)	71.55 (+14.3)

4.2 Implementation Details

We use the default settings to invoke the OpenAI API for text-davinci-003. For ChatGPT, we manually evaluate the results using the corresponding platform and demo websites. For PSSAT [4], we contact the authors and obtain the source code. We follow PSSAT's experiment settings for their upstream work and downstream work. For all the experiments of PSSAT, we conduct the training and testing stages on the A6000 GPU.

4.3 Baselines and Large Language Models

NAT [15] provides two perturbation-aware training methods(data augmentation & stability training) to improve the robustness of the model against perturbations.

PSSAT [4] introduces two MLM-based training strategies to better learn contextual knowledge from perturbed corpus, and utilizes a consistency processing method to filter the generated data.

Text-davinci-003 [2] is the most advanced GPT-3.5 model with 175B parameters.

Table 2. The performance of ChatGPT with different example selection strategies on clean and noisy test sets. The two best methods for each perturbation are marked.

Demonstration	Strategy	Setup	Clean	Character	Word	Sentence			Overall
				Typos	Speech	Paraphrase	Simplification	Verbose	
Instance-oriented level	Random	Clean	**76.92**	58.73	**68.26**	58.61	**74.19**	57.78	65.36
		Augment	68.21	**65.04**	**70.56**	58.82	**73.02**	61.77	**68.34**
		Clean+Augment	71.31	61.01	57.81	53.43	65.03	63.71	62.32
	Retrieve	Clean	56.49	54.84	60.00	45.53	53.84	52.55	54.37
		Augment	61.42	51.67	64.66	51.97	64.00	61.04	60.25
		Clean+Augment	60.32	52.44	60.93	55.82	64.06	44.96	58.24
Entity-oriented level	Random	Clean	74.07	**62.18**	55.39	**75.59**	70.96	**71.75**	**71.55**
		Augment	**78.39**	58.54	58.91	**66.14**	70.40	**64.75**	65.61
		Clean+Augment	70.09	60.19	61.88	64.12	65.50	57.37	63.22
	Retrieve	Clean	53.66	52.89	54.55	54.13	57.37	42.96	53.42
		Augment	70.49	51.72	68.21	55.82	63.86	47.83	61.38
		Clean+Augment	72.58	61.16	65.40	42.46	71.64	46.97	62.84

ChatGPT is created by fine-tuning a GPT-3.5 series model via instruction tuning and reinforcement learning from human feedback (RLHF).

4.4 Main Result

Perturbation Level. Our experimental results are shown in Table 1. LLM performance (ChatGPT, Text-davinci-003) is far behind fintune SOTA (NAT, PSSAT) regardless of whether it is on a clean test set or with perturbations at various levels. This phenomenon may be due to the fact that large models are usually pre-trained on large-scale general training data, which makes it difficult to perform well on specific domain data in zero-shot situations. Further comparison of ChatGPT's performance on clean data and various levels of perturbations shows that large models have the most significant drop in character-level perturbation **Typos** (71.43->40.65) and sentence-level perturbation **Verbose** (71.43->55.56), with no significant changes in **Simplification**. Based on the main experimental table and specific case analysis, we have three explanations:

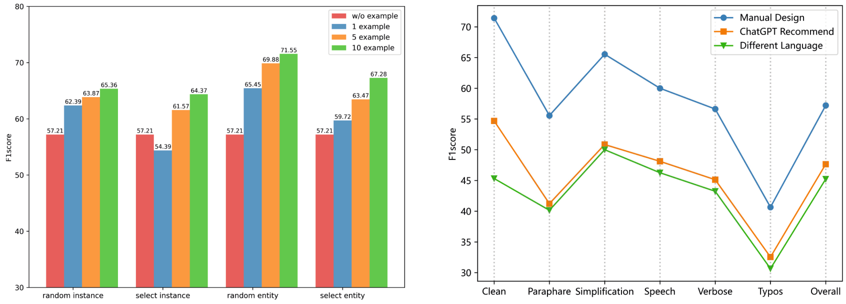
1. Fine-grained perturbation like **Typos** affect the semantics of the entity itself. Although large language models can infer from the context of the sample and have some robustness in recognizing slot entities, the drastic changes in entity semantics can make it difficult to predict the correct entity label.
2. **Verbose** language tends to use redundant expressions to convey the intended meaning, often without explicitly mentioning specific entities. ChatGPT will over-predict entities due to knowledge confusion, which greatly affects its performance in the verbose domain.
3. Despite affecting the semantic information of the original sentence, **simplification** perturbation mitigates the chances of incorrect slot position prediction in ChatGPT by reducing the complexity of input content. Consequently, the model performs relatively better in the Simplification domain.

Demonstration Level. As is shown in Table 2, incorporating entity-oriented and instance-oriented demonstration strategies into ChatGPT indicates a substantial improvement in the overall results (14.34 for Entity and 11.13 for Instance), which demonstrates the effectiveness of our approach. Our evaluation primarily involved analyzing 1) the impact of different types of demonstration, 2) the selection of demonstration, and 3) the distribution of demonstration on the results.

Table 3. The performance of Text-davinci-003 and ChatGPT with best entity-oriented and instance-oriented demonstrations on mixed perturbation.

Methods	Clean	Char	Word	Sen	Char+Word	Word+Sen	Char+Sen	Char+Word+Sen	Overall
		Typos	Speech	AppendIrr	Spe+Typ	Spe+App	Ent+App	Spe+App+Typ	
Text-davince-003	31.24	27.18	23.41	27.48	19.32	19.78	20.73	18.84	24.64
ChatGPT	59.65	42.11	34.83	45.61	27.58	31.03	26.38	26.11	38.18
ChatGPT+Instance level	67.18	48.94	42.25	52.61	34.26	38.79	38.64	30.67	46.58
ChatGPT+Entity level	65.71	47.36	40.37	53.42	36.55	37.35	34.21	29.06	44.27

The Type of Demonstration. Firstly, the entity-level strategy performs better than the instance-level strategy overall, and both strategies show a significant improvement on character-level perturbation (Typos). Specifically, the entity-level strategy exhibits a remarkable improvement on coarse-grained perturbation such as verbose and paraphrase. This is because entity demonstrations provide clear entity boundary information and corresponding labeling, aiding in distinguishing entities from non-entities and greatly reducing entity boundary ambiguity. The instance-level strategy, compared to the entity-level strategy, significantly improves the model's ability to handle Simplify and Speech perturbations. This is due to the diverse context examples provided in instance demonstrations which enhance the model's understanding of domain-specific context information and the distribution of slots, assisting in better reasoning to test samples.



(a) (b)

Fig. 4. (a) The influence of the number of examples in demonstrations. (b) The performance of different types of prompt templates.

The Selection of Demonstration. In most cases, randomly selecting samples proves to be a more effective strategy for both entity-level and instance-level methods, compared to selecting samples based on similarity. The latter may even further reduce the model's performance. One plausible explanation for this finding is that in noisy scenarios, semantic similarity is generally low (¡0.4), making it challenging to ensure consistency with the test sample distribution. By contrast, randomly retrieved samples offer a diverse set of noisy samples that can positively stimulate the context learning ability of large models.

The Distribution of Demonstration. Our experimental results show that, for instance-level models, selecting 10 examples exclusively from the augmented data pool leads to the highest performance on the noisy test dataset. In contrast, for entity-level models, selecting 10 examples from the clean data pool results in the best overall performance on the noisy test dataset. We believe that instance augmented demonstrations enable LLMs to learn the slot information and the correspondence between slots and labels under input perturbations, thereby enhancing the LLMs' ability to understand semantic information in noisy input and increasing their robustness to noise. In contrast, entity demonstrations only provide the correspondence between entities and labels, and selecting examples from the augmented data pool may mislead LLMs to incorrectly identify and classify entities. Furthermore, entity clean demonstrations enable LLMs to learn the correct entity boundaries and the correspondence between entities and labels, thus achieving better performance. However, when we select 10 examples from a mixed data pool of clean and augmented examples, ChatGPT's performance is not as good as when selecting examples exclusively from the clean or augmented data pool. In some cases, it even performs worse than ChatGPT without any examples. This also demonstrates that consistent distribution of inputs in the demonstrations contributes to performance gains during in-context learning.

Mixed Perturbation Experiment. Table 3 shows the performance of different models in facing single and mixed perturbation. Obviously, ChatGPT's performance will drop significantly when facing mixed perturbation, and is lower than all single perturbation results. Although ChatGPT performs better than Text-Davince-003, this improvement becomes very limited in the multi-perturbation scenario. We found that adding entity or instance examples to ChatGPT can effectively improve the model's perturbation robustness. Specifically, the method of adding entity examples improves the overall performance by 22%, while adding instance examples improves it by 21%. Even with the joint interference of three perturbations, our method of selecting examples appropriately can still maintain a 17% improvement in performance, which demonstrates the effectiveness and stability of our proposed approach.

Number of Demonstrations. The Fig. 4(a) shows the impact of the number of examples in demonstrations on the overall performance of chatgpt in single perturbation scenarios. As the number of examples increases, chatgpt's performance improves in both clean and noisy scenarios. Further analysis shows that demonstration selection based on similarity improves with an increase in the number of demos, as the overall semantic similarity between demos and test examples increases and their distribution consistency improves. However, increasing the number of demonstrations through random selection does not improve performance beyond a certain point. This is because as the number of demonstrations increases, their diversity reaches a peak, and additional demonstrations do not provide new semantic information, resulting in stable performance.

Type of Prompts. The Fig. 4(b) shows the impact of different types of prompts on the results. For both the clean and noisy test sets, manually designed prompts perform better than the model-recommended templates. Obviously, when we input detailed descriptions of slot filling task paradigms into LLMs, they can better understand the slot filling task and achieve better performance. When translating the manually designed templates into Chinese, the model's performance also dropped significantly, which verifies our hypothesis that ChatGPT has different levels of understanding of prompts in different languages. Compared to manually constructed templates, ChatGPT's recommended templates show a significant difference, highlighting both the sensitivity of LLMs to templates and the challenge of using LLMs to automate template construction.

5 Conclusion

In this paper, we are the first to comprehensively investigate the effects of input perturbations at different levels on LLMs. We further propose a unified robustness evaluation framework for noisy slot filling to systematically evaluate the dialogue understanding capability of LLMs in diverse input perturbation scenarios. Specifically, we construct a noise evaluation dataset, Noise-LLM, which contains five types of single noise and four types of mixed noise data. Moreover, we utilized a multi-level data augmentation method to construct a candidate data pool, and carefully designed two ways of automatic task demonstration construction strategies with various prompt templates. Experiments result demonstrate that the current open-source LLMs generally have limited ability to counter input perturbations. Our analysis provides empirical guidance for future research.

References

1. Brown, T.B., et al.: Language models are few-shot learners. ArXiv abs/2005.14165 (2020)
2. Brown, T.B., et al.: Language models are few-shot learners (2020)
3. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces (2018)
4. Dong, G., et al.: PSSAT: a perturbed semantic structure awareness transferring method for perturbation-robust slot filling. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 5327–5334. International Committee on Computational Linguistics, Gyeongju (2022). https://aclanthology.org/2022.coling-1.473
5. Gopalakrishnan, K., Hedayatnia, B., Wang, L., Liu, Y., Hakkani-Tur, D.: Are neural open-domain dialog systems robust to speech recognition errors in the dialog history? An empirical study (2020)
6. Gui, T., et al.: TextFlint: unified multilingual robustness evaluation toolkit for natural language processing (2021)
7. Guo, D., et al.: Revisit out-of-vocabulary problem for slot filling: a unified contrastive frameword with multi-level data augmentations. ArXiv abs/2302.13584 (2023)
8. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models (2021)
9. Lazaridou, A., Gribovskaya, E., Stokowiec, W., Grigorev, N.: Internet-augmented language models through few-shot prompting for open-domain question answering. ArXiv abs/2203.05115 (2022)
10. Li, X., et al.: A robust contrastive alignment method for multi-domain text classification. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7827–7831 (2022). https://doi.org/10.1109/ICASSP43922.2022.9747192
11. Li, X., et al.: Generative zero-shot prompt learning for cross-domain slot filling with inverse prompting. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 825–834. Association for Computational Linguistics, Toronto (2023). https://aclanthology.org/2023.findings-acl.52
12. Lin, J., et al.: M6: multi-modality-to-multi-modality multitask mega-transformer for unified pretraining. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD 2021, pp. 3251–3261. Association for Computing Machinery, New York, NY (2021). https://doi.org/10.1145/3447548.3467206
13. Ma, E.: NLP augmentation (2019). https://github.com/makcedward/nlpaug
14. Moradi, M., Samwald, M.: Evaluating the robustness of neural language models to input perturbations. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 1558–1570. Association for Computational Linguistics, Online and Punta Cana (2021). https://doi.org/10.18653/v1/2021.emnlp-main.117, https://aclanthology.org/2021.emnlp-main.117

15. Namysl, M., Behnke, S., Köhler, J.: NAT: noise-aware training for robust neural sequence labeling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1501–1517. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.138, https://aclanthology.org/2020.acl-main.138
16. OpenAI: GPT-4 technical report (2023)
17. Ouyang, L., et al.: Training language models to follow instructions with human feedback (2022)
18. Peng, B., Li, C., Zhang, Z., Zhu, C., Li, J., Gao, J.: RADDLE: an evaluation benchmark and analysis platform for robust task-oriented dialog systems. arXiv preprint arXiv:2012.14666 (2020)
19. Qixiang, G., et al.: Exploiting domain-slot related keywords description for few-shot cross-domain dialogue state tracking. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 2460–2465. Association for Computational Linguistics, Abu Dhabi (2022). https://aclanthology.org/2022.emnlp-main.157
20. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
21. Ruan, W., Nechaev, Y., Chen, L., Su, C., Kiss, I.: Towards an ASR error robust spoken language understanding system. In: Interspeech 2020 (2020). https://www.amazon.science/publications/towards-an-asr-error-robust-spoken-language-understanding-system
22. Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: HuggingGPT: solving AI tasks with ChatGPT and its friends in HuggingFace (2023)
23. Touvron, H., et al.: LLaMA: open and efficient foundation language models (2023)
24. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models (2023)
25. Wu, D., Chen, Y., Ding, L., Tao, D.: Bridging the gap between clean data training and real-world inference for spoken language understanding. arXiv preprint arXiv:2104.06393 (2021)
26. Zeng, W., et al.: Semi-supervised knowledge-grounded pre-training for task-oriented dialog systems. In: Proceedings of the Towards Semi-Supervised and Reinforced Task-Oriented Dialog Systems (SereTOD), pp. 39–47. Association for Computational Linguistics, Abu Dhabi (2022). https://aclanthology.org/2022.seretod-1.6
27. Zhang, Y., et al.: Pay attention to implicit attribute values: a multi-modal generative framework for AVE task. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 13139–13151. Association for Computational Linguistics, Toronto (2023). https://aclanthology.org/2023.findings-acl.831
28. Zhao, G., Dong, G., Shi, Y., Yan, H., Xu, W., Li, S.: Entity-level interaction via heterogeneous graph for multimodal named entity recognition. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 6345–6350. Association for Computational Linguistics, Abu Dhabi (2022). https://aclanthology.org/2022.findings-emnlp.473

Large Language Models are Diverse Role-Players for Summarization Evaluation

Ning Wu, Ming Gong, Linjun Shou, Shining Liang, and Daxin Jiang(✉)

STCA Search and Distribution Group, Microsoft, Beijing, China
{wuning,migon,lisho,shiningliang,djiang}@microsoft.com

Abstract. Text summarization has a wide range of applications in many scenarios. The evaluation of the quality of the generated text is a complex problem. A big challenge to language evaluation is that there is a clear divergence between existing metrics and human evaluation. A document summary's quality can be assessed by human annotators on various criteria, both objective ones like grammar and correctness, and subjective ones like informativeness, succinctness, and appeal. Most of the automatic evaluation methods like BLUE/ROUGE may be not able to adequately capture the above dimensions. In this paper, we propose a new evaluation framework based on LLMs, which provides a comprehensive evaluation framework by comparing generated text and reference text from both objective and subjective aspects. First, we propose to model objective and subjective dimensions of generated text based on roleplayers prompting mechanism. Furthermore, we introduce a context-based prompting mechanism that is able to generate dynamic roleplayer profiles based on input context. Finally, we design a multi-roleplayer prompting technology based on batch prompting and integrate multiple outputs into the final evaluation results. Experimental results on three real datasets for summarization show that our model is highly competitive and has a very high consistency with human annotators.

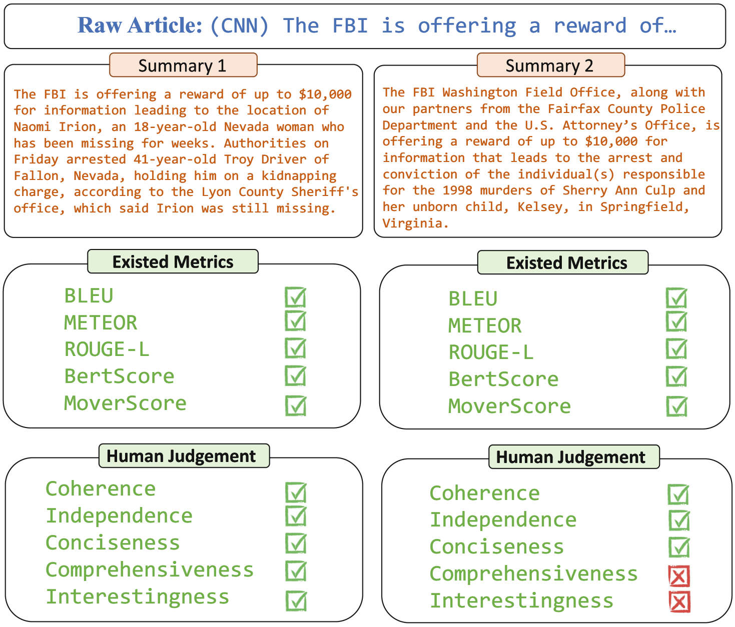
Keywords: Large Language Model · Summarization Evaluation · Role Player

1 Introduction

Text summarization has wide applications in various research and application fields. Recently, some works found that there is a clear gap between existed metrics like BLEU [13], ROUGE, BertScore [20] and human annotations [5,19]. Although typical overlap-based and model-based metrics can capture lexicon level or semantic level similarity between generated text and reference text, specific dimensions like coherence, grammar, and interestingness still can't be captured. As depicted in Fig. 1, the summarization task reveals the inadequacy of traditional metrics such as BLUE/ROUGE: they are unable to reflect the true

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 695–707, 2023.
https://doi.org/10.1007/978-3-031-44693-1_54

quality of the text after reaching a certain level. To achieve consistency between human evaluation and automatic metrics, we encounter two main challenges: 1) How to model objective criteria of evaluation such as coherence and grammar. 2) How to model subjective criteria of evaluation such as interestingness [4,7], comprehensiveness, and usefulness from the standpoint of users. Natural language has various modes of expression for the same concept, so assessing its quality based on a few static criteria is hard.



Summary 1: The FBI is offering a reward of up to $10,000 for information leading to the location of Naomi Irion, an 18-year-old Nevada woman who has been missing for weeks. Authorities on Friday arrested 41-year-old Troy Driver of Fallon, Nevada, holding him on a kidnapping charge, according to the Lyon County Sheriff's office, which said Irion was still missing.

Summary 2: The FBI Washington Field Office, along with our partners from the Fairfax County Police Department and the U.S. Attorney’s Office, is offering a reward of up to $10,000 for information that leads to the arrest and conviction of the individual(s) responsible for the 1998 murders of Sherry Ann Culp and her unborn child, Kelsey, in Springfield, Virginia.

Fig. 1. Two summarizations of CNN News, they are generated by two models (GPT3 [2], T0 [16]), and have similar BLEU and ROUGE metrics, but the second summary is obviously worse than the first one on two more complicated dimensions.

Motivated by the ability of LLMs to handle multi-domains, we investigate how to leverage LLMs for measurement in this paper. Since it is difficult to make LLMs provide a consistent and fair score for the generated text [17], we propose a comparison-based evaluation method to quantify the quality of the generated text, namely DRPE, which stands for Diverse Role-Player for Summarization Generation Evaluation. In particular, we devise a roleplayer-based prompting strategy in this system for objective and subjective dimension measurement. Our method comprises two parts: 1) Static roles construction and dynamic roles prompts generation. 2) A multi-roleplayer framework to conduct a comprehensive evaluation.

For a given generation task, its measurement can be broken down into several dimensions. Typical objective metrics such as coherence and grammar are relatively easy to be agreed upon by most people, so we manually created static roles for each objective dimension of the task. It is expressed as <*Judger types*, *Judger description*>. With a static role, we prompt LLM by asking it to impersonate a real judger based on the judger type and description and then vote for the better option. Furthermore, a comprehensive measurement is usually complex and dynamic. Depending on different cases in a summarization task, different aspects need to be taken into account. Therefore, we propose to dynamically generate some potential users based on the content and let LLMs conduct subjective measurements on behalf of these users. The dynamic roles can be expressed as <*User types*, *User description*>. Lastly, we design a multi-roleplayer framework to eliminate redundant roleplayers and integrate the vote results of multiple roleplayers. Moreover, the multi-roleplayer framework can also enhance the stability of the LLM-based measurement system with relatively low inference costs. Experimental results show that our method significantly surpasses zero-shot LLMs and existing metrics on three typical summarization datasets with human annotations.

2 Related Works

2.1 Large Language Model

Large language model has been found to be capable of few-shot learning [2]. Chain-of-Thought [18] is proposed to empower model reasoning capability for complex tasks. ZeroShot-Cot [8] still shows relatively strong reasoning ability without any examples. Least-to-Most [22] LLM decomposes a complex question into several sub-questions and solves these sub-questions in sequence and finally gives a complete answer to the original question. Recent work [5] discovered that both reference-based and reference-free automatic metrics cannot reliably evaluate zero-shot summaries. In this paper, we mainly explore the capability of LLM to compare the generated text and reference text.

2.2 Existed Metrics

The most widely used metric in machine translation is BLEU [13], which includes several modifications to Exact-P_n. A smoothed variant, SENTBLEU is computed at the sentence level. METEOR [1] computes Exact-P_1 and Exact-R_1 while allowing backing-off from exact unigram matching to matching word stems, synonyms, and paraphrases. ROUGE [9] is a commonly used metric for summarization evaluation. ROUGE-n computes Exact-R_n (usually n = 1, 2), while ROUGE-L is a variant of Exact-R_1 with the numerator replaced by the length of the longest common subsequence. BERTScore [20] computes a similarity score for each token in the candidate sentence with each token in the reference sentence. MoverScore [21] investigates the effectiveness of existing contextualized representations and Earth Mover's Distance [15] for comparing system predictions and reference texts, leading to a new automated evaluation metric that achieves high correlation with human judgments of text quality.

3 Methodology

As discussed in Sect. 1, currently, it forms a gap with human evaluation that automatic metrics for text generation stop at surface similarity (lexicon level or semantic level) which leads to biased perception and evaluation of the text generation capability of LLMs. In this section, we elaborate our proposed measurement framework for text generation primarily includes diversified roleplayers generation and roleplayers-based evaluation (Fig. 2).

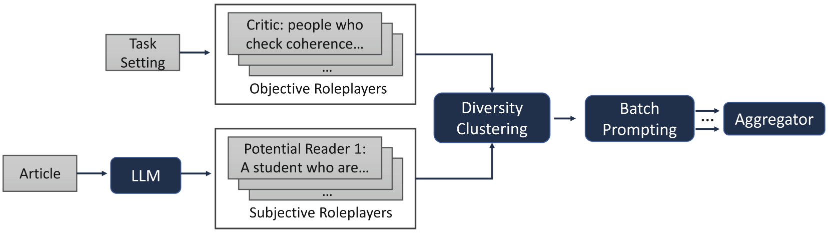
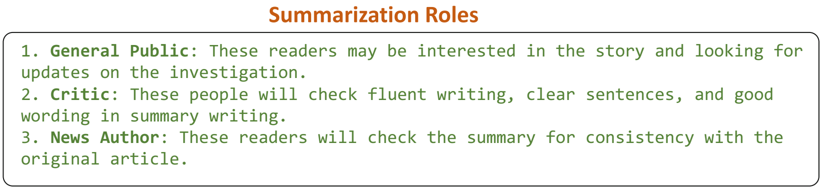


Fig. 2. The overall framework of DRPE. Objective roleplayers are curated manually based on task setting, and subjective roleplayers are dynamically generated by LLMs. After diversity clustering, similar roles are eliminated, and all roles are played by LLMs to compare two candidates in batch prompting. Finally, results from multiple roles are aggregated.

3.1 Diversified RolePlayers Generation

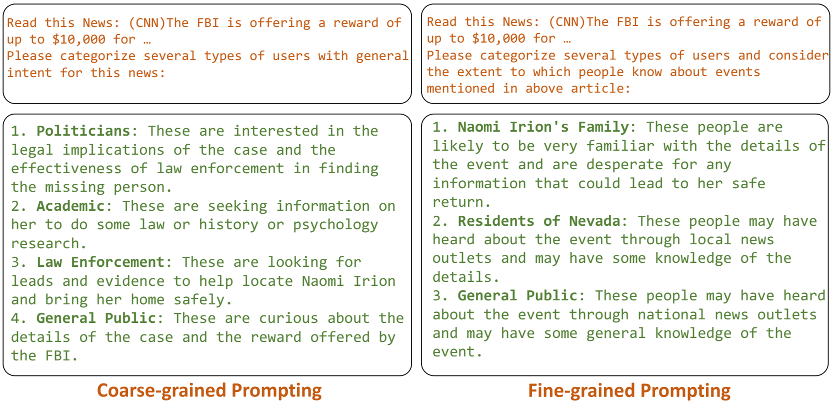
To build a novel framework differing from existing calculation-based automatic metrics, we decompose this task into objective and subjective dimensions and propose an LLM-based measurement method. In our framework, the LLMs act as a judge with a distinctive role to evaluate the text quality in a corresponding dimension and generate its evaluation results. Thus, we need to generate diversified roleplayers for objective and subjective dimensions at first.


Summarization Roles

1. **General Public**: These readers may be interested in the story and looking for updates on the investigation.
2. **Critic**: These people will check fluent writing, clear sentences, and good wording in summary writing.
3. **News Author**: These readers will check the summary for consistency with the original article.

Fig. 3. Three different static roles for summarization task. For different generation tasks, different aspects need to be taken into consideration. A good summary usually requires fluent writing, proper wording and capturing key points of raw article.

Objective RolePlayers. Overlap-based metrics such as BLUE/ROUGE measure lexicon-level consistency with n-grams. Model-based metrics like BERTScore can capture subtle syntax or semantic changes with embeddings similarity, but they have a limitation in evaluating high-quality text from LLMs. The parameter scale of BERT is much smaller than that of LLMs, so it cannot represent LLMs' rich semantics. Consequently in this paper, we take some advanced quality dimensions like fluency, consistency, grammar and coherence into consideration, which were rarely adopted before as they are difficult to be measured accurately. Since different tasks usually require different objective dimensions, and these dimensions are relatively easy to be agreed on by most people, hence we manually curated static objective dimensions for the summarization task and make sure all these dimensions are fundamental and objective. The static objective roles schema is presented below: <*Judger type*, *Judger description*>. Where each *Judger* works on one or multiple specific objective dimensions and *Judger description* breaks down and exposit what specifically the Judger would focus on when evaluating. As shown in Fig. 3, three different objective roles are designed for summarization tasks.



```
Read this News: (CNN)The FBI is offering a reward of
up to $10,000 for …
Please categorize several types of users with general
intent for this news:
```

1. **Politicians**: These are interested in the legal implications of the case and the effectiveness of law enforcement in finding the missing person.
2. **Academic**: These are seeking information on her to do some law or history or psychology research.
3. **Law Enforcement**: These are looking for leads and evidence to help locate Naomi Irion and bring her home safely.
4. **General Public**: These are curious about the details of the case and the reward offered by the FBI.

```
Read this News: (CNN)The FBI is offering a reward of
up to $10,000 for …
Please categorize several types of users and consider
the extent to which people know about events
mentioned in above article:
```

1. **Naomi Irion's Family**: These people are likely to be very familiar with the details of the event and are desperate for any information that could lead to her safe return.
2. **Residents of Nevada**: These people may have heard about the event through local news outlets and may have some knowledge of the details.
3. **General Public**: These people may have heard about the event through national news outlets and may have some general knowledge of the event.

Fig. 4. Coarse-grained and fine-grained prompting mechanism for comprehensive user profiles generation.

Subjective RolePlayers. Text generation, unlike understanding tasks, does not have a perfect golden standard. Human-written material can only offer a high-quality example. Different readers may judge the text according to their own perspectives. For instance, consider the sports news about a renowned athlete in a game,

– For the writing style, ordinary readers expect that it's concise and easy to understand, while journalists would pay attention to its structure and choices of words.

– For the content, causal fans like comprehensive data of the sports player and horizontal comparison with others, while die-hard fans are more eager for in-depth analysis of the sports player through data.

Therefore, we propose to collect subjective evaluations of model-generated text from diverse user perspectives, including whether they think the text is interesting, useful, etc. These dimensions are more abstract and difficult to quantify than objective dimensions which few studies have touched on and addressed to our knowledge. Specifically, we take each generated text as the context and prompt the LLM to generate its potential readers dynamically following the below schema: $<$*User type*, *User description*$>$. Here we design two user role generation prompts. As shown in Fig. 4, the former requires the LLM to consider the most common occupations with most people in the world which is coarse-grained and the latter aims to categorize people based on their familiarity with the text topics which is fine-grained. We merge objective judgers and subjective users generated by two kinds of prompting mechanisms as multi-role players for the next process. Considering that there may exist duplicate or similar users, we propose to conduct diversity clustering to improve measurement performance and reduce inference costs. First, each roleplayer type and its description are concatenated as the input of Sentence-BERT [14] to obtain the representation. Next, we use the k-means algorithm to cluster roleplayers, and those closest to each cluster center are kept. Finally, the chosen role players will be leveraged for text evaluation.

3.2 RolePlayer-Based Evaluation

To mitigate the discrepancy between the human evaluation and automatic metrics, we propose to leverage the roleplayers as crowdsourcing voters to compare the summaries from multi-dimensions. Besides the static roleplayers that scrutinize the objective dimensions including grammar, fluency, coherence, etc., the dynamic roleplayers are generated according to the current article and simulate the psychology and behavior of the article readers (roles) to convey their subjective feelings. It's expected that our method could achieve higher consistency with human evaluation than existing automatic metrics focusing on surface similarity.

Evaluation of RolePlayer. Given the article A, we posit the reference summary generated by humans as S, and the candidate summary generated by models as $\hat{S}$, respectively. To evaluate, all the roleplayers perform pair-wise comparison as Fig. 5, since point-wise comparisons are inconsistent across different samples and list-wise comparisons are not stable. By our prompting, the LLMs play a specific role and output its analysis of two summaries, finally voting which summary is of better quality. In more detail, we parse and quantify the comparison result as $\hat{a}$:

$$\hat{a} = \begin{cases} 1 & \text{If voting is candidate summary } \hat{S} \\ 0 & \text{If voting is reference summary } S \end{cases} \tag{1}$$

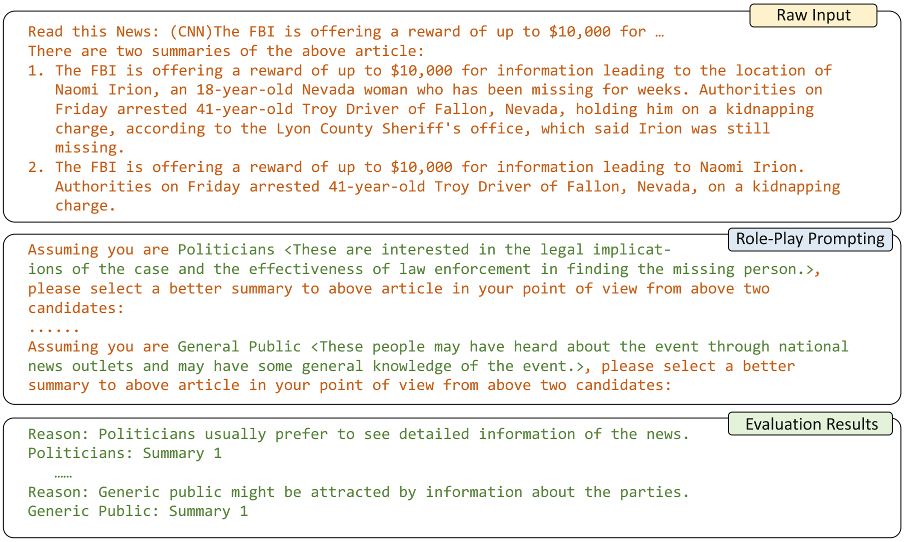
Assuming the DRPE score for generated text $\hat{S}$ is $\mathrm{DRPE}(\hat{S}|A, S)$, it could be obtained by modeling the joint probability of reason and voting result as below:

$$\mathrm{DRPE}(\hat{S}|A, S) = \mathbb{1}(\hat{a} = 1)P(\hat{a}, \boldsymbol{r}|\boldsymbol{p}, A, S, \hat{S}, R) \tag{2}$$

where $\boldsymbol{r}$ is the comparison reason, $\boldsymbol{p}$ represents the prompt used here, $\hat{a}$ is voting result from LLMs, A is raw article and R is the role. To compute $P(\hat{a}, \boldsymbol{a})$, similar to [17], we leverage the confidence when roleplayer generates its voting normalized by the output length [2],

$$\begin{aligned} &P(\hat{a}, \boldsymbol{r}|\boldsymbol{p}, A, S, \hat{S}, R) \\ &= \exp^{\frac{1}{K}\sum_{k=1}^{K}\log P(t_k|\boldsymbol{p}, A, S, \hat{S}, R, t_1, \ldots, t_{k-1})} \end{aligned} \tag{3}$$

where $\log P(t_k|\boldsymbol{p}, A, S, \hat{S}, R, t_1, \ldots, t_{k-1})$ is the log probability of k-th token t_k in $\boldsymbol{r}$ and K is number of tokens.



Read this News: (CNN)The FBI is offering a reward of up to $10,000 for …
There are two summaries of the above article:
1. The FBI is offering a reward of up to $10,000 for information leading to the location of Naomi Irion, an 18-year-old Nevada woman who has been missing for weeks. Authorities on Friday arrested 41-year-old Troy Driver of Fallon, Nevada, holding him on a kidnapping charge, according to the Lyon County Sheriff's office, which said Irion was still missing.
2. The FBI is offering a reward of up to $10,000 for information leading to Naomi Irion. Authorities on Friday arrested 41-year-old Troy Driver of Fallon, Nevada, on a kidnapping charge.

Assuming you are Politicians <These are interested in the legal implications of the case and the effectiveness of law enforcement in finding the missing person.>, please select a better summary to above article in your point of view from above two candidates:
......
Assuming you are General Public <These people may have heard about the event through national news outlets and may have some general knowledge of the event.>, please select a better summary to above article in your point of view from above two candidates:

Reason: Politicians usually prefer to see detailed information of the news.
Politicians: Summary 1
……
Reason: Generic public might be attracted by information about the parties.
Generic Public: Summary 1

Fig. 5. Compare generated summary and reference summary by multiple roleplayers with batch prompting.

Batch Prompting. To efficiently get comparison results for the summary pair (S, $\hat{S}$) from multiple roleplayers, we design a multi-roleplayer framework based on batch prompting to measure by both objective and subjective metrics in one-off inference. The different metrics are reflected by the objective and subjective roleplayers generated and clustered in Sect. 3.1. As shown in Fig. 5, first, all the roleplayers are prompted to give votes for (S, $\hat{S}$) with respect to A in their point of view, i.e., which summary could better describe the article, and tell us the reasons. Then we aggregate the results to parse $\hat{a} = \{\hat{a}_j\}_{j=1}^N$ where N is the number of roleplayers. According to Eq. 2, the multi-roleplayer DRPE score by batch prompting can be formulated as below:

$$\begin{aligned} \mathrm{DRPE}(\hat{S}|A,S) &= P(\hat{a}|\boldsymbol{p},A,S,\hat{S}) \\ &= \sum_{j=1}^{N} \mathbb{1}(\hat{a}_j = 1) P(\hat{a}_j, \boldsymbol{r}|\boldsymbol{p},A,S,\hat{S},R_j) \end{aligned} \tag{4}$$

where R_j denotes j-th roles. Compared with Self-Consistency CoT [17], our framework decouples the answer (comparison result) and reasoning path (comparison reason), and brings higher inference efficiency as Self-Consistency CoT needs to generate each candidating answer separately while our method generates all voting results with once inference.

4 Experiments

4.1 Experiments Setting

Datasets. • **CNN2022** [6,11]: contains reference summaries that are approximately 3–4 sentences long. Summaries in this dataset are highly extractive and lead-biased. We use human study data on 100 recent articles from CNN, collected between March 1, 2022, and June 31, 2022. Each article is annotated by at least three judgers, and they are asked to choose the best and worst summaries from three candidates. To reduce noises, we only keep the best summary with at least 2 out of 3 votes and worst summary with at least 2 out of 3 votes. Finally, we obtain 81 best and worst summaries as candidate summaries with a piece of corresponding news. Finally, we use GPT-3 of the text-DaVinci-003 to generate reference summarization, and finally use our method to compare the candidate summary.

• **BBC2022** [12]: contains 1 sentence summaries of BBC news articles. In this dataset, references summaries, and consequently generated summaries from fine-tuned models are highly abstractive. We use human study data on 100 recent articles from BBC, collected between March 1, 2022, and June 31, 2022. We also take a similar preprocessing strategy like CNN2022 on this dataset.

• **SummEval** [3]: contains a large and diverse collection of human judgments of model-generated summaries on the CNN/Daily Mail dataset annotated by both expert judges and crowd-source workers. For each news, we select two worst summaries and two best summaries according to their average scores on four dimensions (coherence, consistency, fluency and relevance) labeled by experts. Finally, regarding one hundred news and corresponding reference summary in SummEval, we obtain 400 candidate summaries.

Metrics. To measure the consistency between various metrics and human annotations, we follow the WMT18 [10] standard practice and use absolute Pearson correlation $|\rho|$ to evaluate metric quality.

Baselines. Automatic metrics proposed for summarization evaluation can be broadly divided into two categories: (1) overlap-based metrics, specifically ROUGE METEOR and BLEU, and (2) similarity-based metrics that compute the similarity between embeddings representations of generated and reference summaries. Specifically, we report BERTScore and MoverScore. For LLMScore, we carefully design prompts for LLM, and directly use it to predict a better passage.

Implementation. We use the public GPT-3 [2] of the text-DaVinci-003 version with 175B parameters from OpenAI for the LLMs implementation and use greedy decoding for inference with the temperature set to 0. We select this LLM because it has relatively good capability among public LLMs. Especially, we use three roles *General Public*, *Critic*, and *News Author* which are described in Fig. 3 as objective roleplayers in our DRPE, and we prompt the model to generate 4 dynamic roles for each case.

4.2 Results

Table 1. Pearson correlation between several automatic metrics and human annotation. We bold the highest numbers for each dataset, and use AVG to denote the average scores on the three datasets. Results of GPT-D3 and DRPE are averaged over five runs with slight changing in prompts.

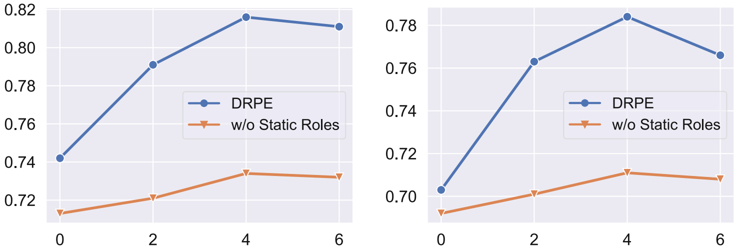
Type	Method	CNN2022	SummEval	BBC2022	AVG
Overlap	ROUGE-1	0.466	0.431	0.469	0.461
	ROUGE-2	0.437	0.354	0.443	0.411
	ROUGE-L	0.422	0.322	0.436	0.393
	BLEU	0.475	0.372	0.502	0.450
	METEOR	0.514	0.473	0.561	0.516
Similarity	BERTScore	0.554	0.455	0.568	0.526
	MoverScore	0.456	0.385	0.442	0.428
LLM	GPT-D3	0.713	0.503	0.692	0.636
	DRPE	**0.816**	**0.683**	**0.784**	**0.761**

Comparison with Existed Metrics. Tables 1 shows Pearson correlation to human judgments. We observe that typical overlap-based metrics generally performed badly and relevance-based metrics also underperformed. The simplest LLM-based method has a consistently better performance than BERTScore. Two types of LLM-based methods, GPT-D3, and DRPEScore have a clear gap between the other two methods. Especially, DRPEScore consistently performs better than GPT-D3.

Table 2. Pearson correlation between several automatic metrics and human annotation. AVG denotes the average scores on the three datasets.

Method	CNN2022	SummEval	BBC2022	Avg
DRPE	0.816	0.683	0.784	0.761
w/o Batch Inferring	0.822	0.672	0.766	0.753
w/o Clustering	0.782	0.665	0.751	0.733
w/o Dynamic Roles	0.742	0.669	0.703	0.705
w/o Static Roles	0.734	0.604	0.711	0.683

Ablation Study. We conducted an ablation study on DRPE to gain insights into the detailed method design. We prepare four variants of our method: (1) w/o Batch Inferring denotes without the batch prompting, each role is inferred alone; (2) w/o Clustering denotes without clustering mechansim; (3) w/o Dynamic Roles denotes without dynamic roles generation; (4) w/o Static Roles denotes without the human designed static roles. Table 2 presents all comparison results of the four variants. As we can see, the performance rank on three datasets can be given as: w/o Static Roles < w/o Dynamic Roles < w/o Clustering < w/o Batch Inferring < DRPE. These results indicate that all components are essential to improve performance. And we can also find that batch inferring is able to save lots of inference tokens without losing performance.

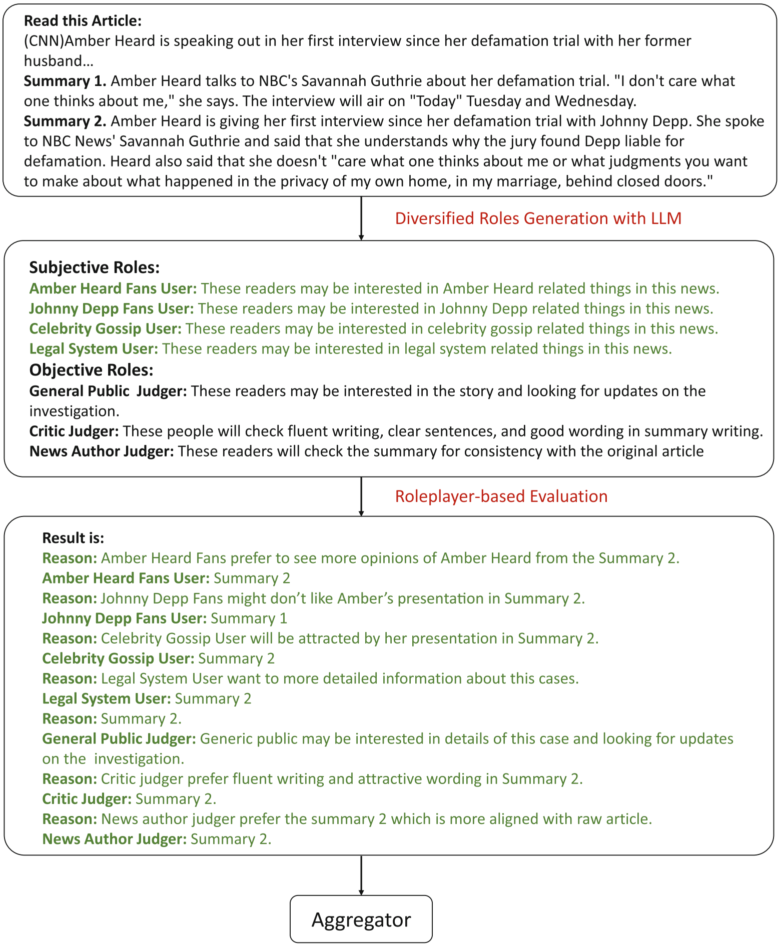


(a) Roles number k w.r.t correlation on CNN2022 dataset.

(b) Roles number k w.r.t correlation on BBC2022 dataset.

Fig. 6. Effect of role number on model performance.

Effects of Hyperparameters. We test DRPEScore and w/o Static Roles with different subjective role numbers k in [0, 2, 4, 6] on two datasets. Figure 6 shows the experimental results. When k increases from 0 to 6, the experimental indicators first increase and then slightly decrease on the CNN2022 dataset, when $k = 4$, the correlation to human evaluation achieves a peak. On the BBC2022 dataset, experimental results are similar, and more roles (6) don't bring improvements compared to fewer roles (4).



Read this Article:
(CNN)Amber Heard is speaking out in her first interview since her defamation trial with her former husband...
Summary 1. Amber Heard talks to NBC's Savannah Guthrie about her defamation trial. "I don't care what one thinks about me," she says. The interview will air on "Today" Tuesday and Wednesday.
Summary 2. Amber Heard is giving her first interview since her defamation trial with Johnny Depp. She spoke to NBC News' Savannah Guthrie and said that she understands why the jury found Depp liable for defamation. Heard also said that she doesn't "care what one thinks about me or what judgments you want to make about what happened in the privacy of my own home, in my marriage, behind closed doors."

Subjective Roles:
Amber Heard Fans User: These readers may be interested in Amber Heard related things in this news.
Johnny Depp Fans User: These readers may be interested in Johnny Depp related things in this news.
Celebrity Gossip User: These readers may be interested in celebrity gossip related things in this news.
Legal System User: These readers may be interested in legal system related things in this news.
Objective Roles:
General Public Judger: These readers may be interested in the story and looking for updates on the investigation.
Critic Judger: These people will check fluent writing, clear sentences, and good wording in summary writing.
News Author Judger: These readers will check the summary for consistency with the original article

Result is:
Reason: Amber Heard Fans prefer to see more opinions of Amber Heard from the Summary 2.
Amber Heard Fans User: Summary 2
Reason: Johnny Depp Fans might don't like Amber's presentation in Summary 2.
Johnny Depp Fans User: Summary 1
Reason: Celebrity Gossip User will be attracted by her presentation in Summary 2.
Celebrity Gossip User: Summary 2
Reason: Legal System User want to more detailed information about this cases.
Legal System User: Summary 2
Reason: Summary 2.
General Public Judger: Generic public may be interested in details of this case and looking for updates on the investigation.
Reason: Critic judger prefer fluent writing and attractive wording in Summary 2.
Critic Judger: Summary 2.
Reason: News author judger prefer the summary 2 which is more aligned with raw article.
News Author Judger: Summary 2.

Fig. 7. Evaluation procedure for two summaries given a piece of news. We use the green font to represent the content generated by the model. With suitable prompting, the LLM is able to generate relevant roles and generate judgment for these roles. (Color figure online)

Qualitative Analysis. We have previously demonstrated the effectiveness of our model on two summarization tasks. In this section, we conduct a qualitative analysis to explain why DRPEScore can achieve good performance. Figure 7 shows an example of our model's evaluation process. Given a news article about the legal dispute between Amber Heard and Johnny Depp, our model has to select a better summary from two candidates. First, we generate several subjective roles based on the news content, such as *Amber Heard Fans User*, *Johnny Depp Fans User*, *Celebrity Gossip User*, and *Legal System User*. These roles are representative of different perspectives and preferences that can be captured by LLMs. Second, we use LLMs to simulate each user and judger and compare the two summaries. We employ a batch prompting mechanism to accelerate the inference procedure. Notably, LLMs predict that the *Johnny Depp Fans User*, who might have a negative attitude towards *Amber Heard Fans User*, will favor summary 1.

5 Conclusion

We propose DRPE, a new comparison-based method for evaluating generated text against gold standard references. Our DRPE is designed to be simple and task-agnostic. Our experiments illustrate that DRPE could provide a human-like ability to conduct a comprehensive evaluation, especially on challenging long text generation like summarization tasks. In future work, we look forward to exploring the capabilities of LLMs as judgers on more text evaluation tasks and reducing the computation cost.

References

1. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation AND/OR Summarization, pp. 65–72 (2005)
2. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
3. Fabbri, A.R., Kryściński, W., McCann, B., Xiong, C., Socher, R., Radev, D.: SummEval: re-evaluating summarization evaluation. TACL **9**, 391–409 (2021)
4. Gao, J., Pantel, P., Gamon, M., He, X., Deng, L.: Modeling interestingness with deep neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2–13 (2014)
5. Goyal, T., Li, J.J., Durrett, G.: News summarization and evaluation in the era of GPT-3. arXiv preprint arXiv:2209.12356 (2022)
6. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: NIPS, vol. 28 (2015)
7. Hidi, S., Baird, W.: Interestingness–a neglected variable in discourse processing. Cogn. Sci. **10**(2), 179–194 (1986)
8. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. arXiv preprint arXiv:2205.11916 (2022)
9. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
10. Ma, Q., Bojar, O., Graham, Y.: Results of the WMT18 metrics shared task: both characters and embeddings achieve good performance. In: Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pp. 671–688 (2018)

11. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023 (2016)
12. Narayan, S., Cohen, S.B., Lapata, M.: Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745 (2018)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
14. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: EMNLP, pp. 3982–3992 (2019)
15. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vision **40**(2), 99 (2000)
16. Sanh, V., et al.: Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207 (2021)
17. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
18. Wei, J., et al.: Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903 (2022)
19. Yuan, X., et al.: Selecting better samples from pre-trained LLMs: a case study on question generation. arXiv preprint arXiv:2209.11000 (2022)
20. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT. arXiv preprint arXiv:1904.09675 (2019)
21. Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C.M., Eger, S.: MoverScore: text generation evaluating with contextualized embeddings and earth mover distance. arXiv preprint arXiv:1909.02622 (2019)
22. Zhou, D., et al.: Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv:2205.10625 (2022)

COSYWA: Enhancing Semantic Integrity in Watermarking Natural Language Generation

Junjie Fang[1], Zhixing Tan[2], and Xiaodong Shi[1,3](✉)

[1] Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen, China
fangjj@stu.xmu.edu.cn, mandel@xmu.edu.cn

[2] Zhongguancun Laboratory, Beijing, People's Republic of China
zxtan@zgclab.edu.cn

[3] Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, Xiamen, China

Abstract. With the increasing use of natural language generation (NLG) models, there is a growing need to differentiate between machine-generated text and natural language text. One promising approach is watermarking, which can help identify machine-generated text and protect against risks such as spam emails and academic dishonesty. However, existing watermarking methods can significantly affect the semantic meaning of the text, creating a need for more effective techniques that maintain semantic integrity. In this paper, we propose a novel watermarking method called **CO**ntextual **SY**nonym **WA**termarking (**COSYWA**) that embeds watermarks in text using a Masked Language Model (MLM) without significantly impairing its semantics. Specifically, we use post-processing to embed watermarks in the output of an NLG model. We generate a context-based synonym set using an MLM model to embed watermark information and use statistical hypothesis testing to detect the existence of watermarking. Our experimental results show that COSYWA significantly enhances the text's capacity to maintain its original meaning while effectively embedding a watermark, making it a promising approach for protecting against misinformation in NLG.

Keywords: Natural Language Generation · Watermarking · Contextual Synonym

1 Introduction

Large Natural Language Generation (NLG) models, such as ChatGPT [19], have demonstrated impressive performance on various downstream tasks and can generate text that closely resembles natural language. However, with the increasing prevalence of NLG models, the risk of their misuse is also rising, including

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 708–720, 2023.
https://doi.org/10.1007/978-3-031-44693-1_55

spreading fake news and information [24,25], perpetrating phishing scams [2,10], engaging in academic dishonesty [8,26], and generating spamming emails [14]. Detecting machine-generated text has become essential in curbing the misuse of NLG models [3,7]. Language model watermarking is an effective method for distinguishing machine-generated text from natural text [13]. It can embed hidden information into text and can be detected using corresponding watermark detection algorithms to determine whether the current text contains a watermark, i.e., whether it is machine-generated.

There are two main types of natural language watermarking: decoding-based and post-processing watermarking. Decoding-based watermarking modifies the predicted probability distribution during the decoding process of a language generation model to embed the watermark [13]. However, our experiments reveal that this approach can significantly alter the semantic meaning of the text, which may adversely affect the performance of tasks such as translation and summarization. Post-processing watermarking embeds watermark information by processing a given text, primarily through adjusting the synonyms [11,12] or spelling [12] of the text. This approach can preserve semantic information better but faces three primary challenges. Firstly, the process of designing synonym replacement watermark rules is complicated and requires multiple mechanisms to confirm acceptability, such as verifying whether they conform to the current context [5]. Secondly, due to the complexity of natural language, words are only synonymous in specific contexts and are not exchangeable in all situations [6]. Finally, current synonym-based methods employ fixed watermarking rules, which may systematically modify the text's statistical information, compromising the confidentiality of the watermark and enabling adversaries to automatically detect and remove it [11].

To ensure semantic consistency between pre- and post-watermarked text, we develop a robust watermarking system called **CO**ntextual **SY**nonym **WA**termark (COSYWA), rooted in post-processing watermarking. To mitigate the challenge of accessing external synonym knowledge bases, inconsistencies in semantic meanings of static synonyms, and the scarcity of diverse replacement rules, we use a Masked Language Model (MLM) that can dynamically construct a candidate synonym set for flexible and context-based synonym replacement. The synonym replacement strategy using the MLM model empowers our watermarking approach to form a synonym set better aligned with contextual semantics, thus maintaining higher semantic consistency between the original and watermarked sentences. Furthermore, the MLM model brings enhanced flexibility and variability to the watermark replacement rules, thereby increasing their resistance to detection through statistical analysis methods. To improve robustness, we integrate a synonym approach with vocabulary encoding, where we substitute adjectives in the text with words defined in the vocabulary. During the watermark detection phase, we employ a model-free watermark verification approach.

We conduct experiments to validate the effectiveness of COSYWA in translation and summarization tasks. Results indicate that our approach outperforms the baseline watermarking methods in maintaining the lexical and semantic equivalence of the watermarked text. Additionally, the watermark strength is much higher for translation and summarization tasks compared to decoding-based watermarks and synonym-based watermarks.

2 Related Work

Natural language watermarking techniques can be broadly classified into two types: decoding-based watermarking and post-processing watermarking.

2.1 Decoding-Based Watermarking

Kirchenbauer et al. [13] first propose a watermarking method during the inference process. During the word generation process, a hash code is created based on the previously generated word, and it is used to seed a random number generator, which randomly divides the entire vocabulary into "green lists" and "red lists" of equal size. The next word is then generated from the green list, allowing the watermark to be embedded in each generated word. To detect the watermark, a third-party detector can reproduce the hash function and random number generator to copy the red list for each word and count the number of violations of the red list rule to verify the authenticity of the text. The probability of generating N-marked tokens in the natural text without violating the red list rule is only $1/2^N$, even for text fragments with only a few words. This feature makes the COSYWA method transferable, as edit-based methods only modify some words in the text. However, decoding-based watermarking during controlled sampling may have a great impact on the semantics of the resulting text.

2.2 Post-processing Watermarking

He et al. [12] propose a synonymy-based watermarking method by replacing certain words in the text with their synonyms. However, it has been found that synonym-based methods that use fixed rule-based substitutions may systematically alter the statistical information of the text, which could compromise the confidentiality of the watermark and enable attackers to automatically detect and remove it [11]. Furthermore, early synonym substitution methods require external knowledge bases, and the construction of such watermark rules can be cumbersome. Additionally, due to the complexity of linguistic phenomena such as part-of-speech, polysemy, and context semantics, many synonyms only satisfy synonymy conditions in specific contexts and are not applicable in all cases [6]. We improve the semantic preservation of watermarked text by extending existing synonym-based watermark methods [11,12]. Instead of relying on a static external synonym database, we used an MLM to dynamically generate synonyms during the watermarking process. By considering the semantic context of the original words when selecting synonyms, we ensure that different synonyms are chosen for the same word in different semantic contexts. We also adopt the vocabulary partitioning technique from decoding-based watermark [13] to enhance the robustness of synonym-based watermarks.

3 Methodology

In this section, we introduce a novel post-processing watermarking method called COSYWA. COSYWA first identifies appropriate watermark words that can be replaced without significantly altering the model's semantic understanding of the text. It then replaces these words with semantically similar words that carry the watermark information, using the MLM. For watermark detection, COSYWA applies hypothesis testing to determine the likelihood of a given sentence containing the watermark information.

3.1 Part-of-Speech

For a given sentence $S = [w_1, w_2, \ldots, w_N]$, where w_i represents the i-th word in the sentence and N is the number of words in the sentence, we label each word in the sentence using a Part-of-Speech tagger to obtain the corresponding part-of-speech, i.e., $POS = [pos_1, pos_2, \ldots, pos_N]$, where pos_i represents the part of speech of the i-th word. We select adjective words as watermark words following the previous work [12].

3.2 Word Synonym Set Construction via BERT

Previous approaches rely on external knowledge bases like WordNet to construct the replacement space for watermark words [6,11,12,28], which may lack context awareness, leading to compromised fluency control and semantic consistency. To mitigate this, we propose using BERT to construct the replacement word space, as it generates relatively fluent and grammatically correct sentences while preserving most of the semantic information. Furthermore, unlike WordNet-based approaches, BERT is context-aware and dynamically searches for replacement words, instead of simply substituting with synonyms.

To predict possible replacements for a selected word w, we utilize BERT's MLM. Unlike the standard MLM setting, we directly use the original sequence as the input and do not replace the selected word w to the `[MASK]` token. This approach generates more semantically consistent and fluent alternatives [31]. For instance, consider the sequence "I like apples". If we mask the word "apples", the MLM may predict "I like bananas" with equal fluency, making it difficult to predict the original word "apples".

Specifically, we input the given text S into the BERT model, obtaining the representation of each token $P = [p_1, p_2, \ldots, p_N]$, where N is the total number of tokens in the current sentence. For the token replacement process, we adopt the method from [16].

3.3 Word Replacement Strategy

During the construction of the replacement word space, we first generate a hash code for each word in the vocabulary and use it as a seed for a random number generator. This seed randomly selects γ proportion of words in the entire vocabulary as the green list and the rest as the red list. Each word in the vocabulary thus corresponds to a unique green and red list.

We construct the replacement word space for watermark words by hashing the words that precede the watermark word to determine the green list of vocabulary. Then, we select the top K words with the highest probabilities from the green list and measure the change in the BERTScore of the sentence after replacement. Finally, we select the word with the highest BERTScore among the same part of speech for replacement. This way, the watermark is embedded in each generated word. The complete algorithm is shown in Algorithm 1.

Algorithm 1. Word Replacement with Green List

```
Input: sentence, $w_1$, ..., $w_N$; part-of-speech of sentence, $pos_1$, ..., $pos_N$
Apply the language model to all token $w_1$, ..., $w_N$ to get a probability vector $p_1$, ..., $p_N$ over the vocabulary.
choose watermark part-of-speech $pos_w$
for $t = 0, 1, \ldots$ do
    if $pos_t == pos_w$ then
        compute a hash of token $w_{t-1}$ and utilize it as a seed for a random number generator.
        With this seed, assign random colors to the vocabulary, creating a "green list" $G$ and a "red list" $R$.
        When sampling the token $w_t$ from $p_t$, ensure that $w_t$ is selected from the green list $G$ and avoid choosing any tokens from the red list $R$.
    end if
end for
```

Figure 1 illustrate an example for COSYWA. We first employ the Spacy POS tagger to assign a part-of-speech tag for each word in the given sentence, and identify the watermark word that meets our criteria, which in Fig. 1 is the adjective "attractive". Next, we generate a synonym space for "attractive" by inputting the original sentence into the MLM, and then use the green list associated with the word "is" to color the probability distribution of the MLM's prediction for "attractive". If the current adjective is not included in the green list, we choose a word from the prediction results that meet the replacement conditions and is part of the green list. This word is then used to replace the original adjective where the word "handsome" is used to replace "attractive".

3.4 Watermark Detection

Although generating watermarked text requires the use of an additional language model, detecting the watermark does not. Detectors familiar with the hash function and random number generator can regenerate the green lists for each word and count the number of violations of the green list rule. We can

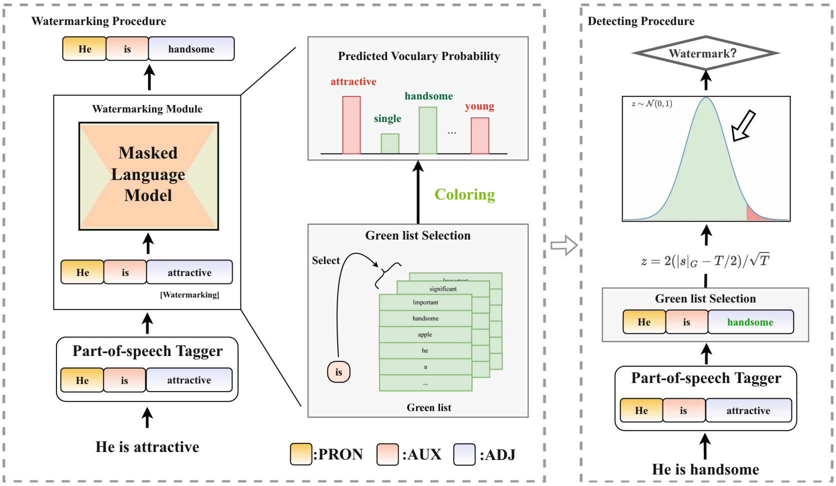


Fig. 1. An running example for COSYWA. COSYWA first embeds a watermark into a given sentence using the MLM (left) and then detects a suspicious sentence by testing the null hypothesis (right).

detect the watermark by testing the following null hypothesis: *The text sequence is generated without the watermark rule.*

A robust method for detecting watermarks involves using a one-proportion z-test to evaluate the null hypothesis. The z-statistic for this test is

$$z = (|s|_G - T\gamma)/\sqrt{T\gamma(1-\gamma)} \tag{1}$$

Language models consist of a vocabulary V, which includes a collection of words. We create a green list G with a size of $\gamma|V|$. For a given sentence, the number of tokens from the green list is represented as $|s|_G$, and T corresponds to the number of watermark words. We can reject the null hypothesis and detect the watermark if z is above a chosen threshold. Assuming a threshold of $z > 4$, the probability of a false positive is 3×10^{-5}, corresponding to a one-tailed p-value for $z > 4$.

4 Experiments

4.1 Settings

Tasks and Datasets. We evaluate the effectiveness of COSYWA on machine translation and document summarization tasks.

Machine Translation. We use the WMT14 German to English (De-En) translation dataset [4] as a testing platform to explore performance in translation scenarios. We evaluate translation quality using BLEU [20] and BERTScore [30], selecting parts of the dataset where the text length is greater than 100. We employ `facebook/wmt-19-de-en` [18] as the translation model.

Summarization. We use the CNN/DM [23] dataset to evaluate the performance of the proposed method on document summarization tasks. The dataset aims to summarize news articles into information summaries. We use the bart-large-cnn [15] for document summarization and employed Rouge-L [17] and BERTScore [30] as evaluation metrics for summary quality.

Models. We utilize BERT [9] as the masked LM for the proposed edit-based method, specifically the BERTBase-uncased model and the transformers package from Hugging Face [29]. As for the pre-trained language models for each specific task, including `Facebook/wmt-19-de-en` [18], and `bart-large-cnn` [15], we directly utilize the pre-trained models provided by Hugging Face [29].

Implementation Details. We generate the output of each task model and then embed the watermark to the generated results. To improve the accuracy of the POS tagger, we remove line breaks from the output. Since we use the bert-base-uncased model in our experiments, we convert all output data to lowercase to test its performance. After preprocessing, we evaluate the model's effectiveness. We implement COSYWA using the PyTorch backend of the Hugging Face library [29]. As described in Sect. 3, we use the torch random number generator and the previous token to generate the green list. To evaluate the watermark method's performance, we follow previous works [1,27] and use 200 random sentence pairs from each task dataset as our test set. In the experiment, we set the candidate set K to 96 and the green list proportion γ to 0.25.

Baselines. We compare our method with two baselines [12,13]. Kirchenbauer et al. [13] propose adding watermarks during the decoding process by perturbing the probability distribution to embed watermark information. He et al. [12] design a synonym-based watermarking method that replaces all watermarked words with synonyms.

4.2 Results

Table 1 illustrates the impact of Kirchaenbauer et al.'s method on task performance. The inclusion of hard watermarks successfully embeds them into the text but significantly affects task performance. Furthermore, the inclusion of a soft watermark has a substantial positive effect on performance. However, along with performance enhancement, it also presents a significant challenge: the intricate task of embedding the soft watermark into low-entropy text, such as translation tasks and document summarization tasks with standardized answers. The low Z-score value of the soft watermark is evident from the data presented in Table 1. It is important to note that the Z-score plays a critical role in watermarking tasks as it determines the success of watermark embedding and detectability.

Regarding He et al.'s watermarking method [12], while it has little effect on semantics and task performance, it is mainly applied for protecting model intellectual property. This method requires small datasets for watermark detection.

Table 1. Performance of different watermarking approaches on WMT14 and CNN/DM. BS means BERTScore. RL means Rouge-L. Calculating the Z-score differs among various watermarking methods based on the number of watermark words, denoted as T like text sequence in [13], the count of adjectives in [12], and COSYWA.

Method	WMT14			CNN/DM		
	Z-score	BLEU	BS	Z-score	RL	BS
Kirchenbauer et al. [13]						
w/o watermark	0.31	12.02	76.73	−0.76	13.38	78.36
hard watermark	4.62	0.63	73.44	4.14	4.44	74.7
soft watermark	1.93	10.85	76.9	0.85	12.24	78.13
He et al. [12]						
w/o watermark	0.39	12.02	76.73	−0.72	13.38	78.36
watemark	0.44	11.61	76.55	−0.14	13.26	78.4
our method						
w/o watermark	−0.15	12.02	76.73	0.69	13.38	78.36
watermark	3.44	10.46	76.57	4.82	12.92	78.31

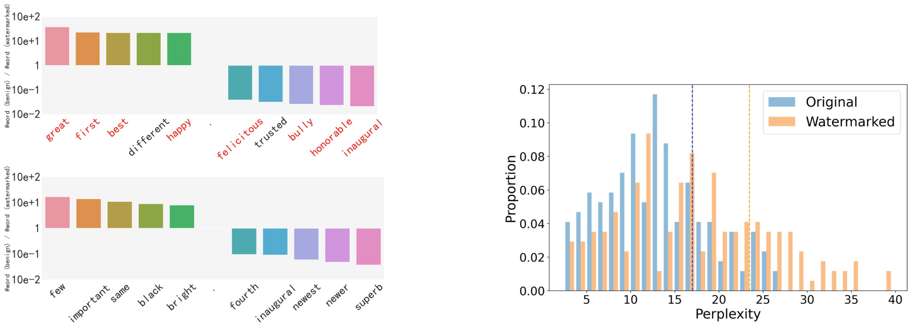
Nevertheless, the experiments indicate its unsuitability for embedding detectable watermarks in sentence-level data.

COSYWA is more easily detectable while maintaining model performance. COSYWA demonstrates better preservation of lexical and semantic features in tasks with answer restrictions. For translation and document summarization tasks, COSYWA achieves better maintenance of BLEU and ROUGE-L performance at the same watermark strength. In essence, COSYWA combines the advantages of hard and soft watermarks by retaining semantic integrity like soft watermarks and providing strong watermarking capabilities like hard watermarks. Furthermore, we employed BERTScore [30] to evaluate different watermarking methods, utilizing context embeddings for assessing semantic equivalence. Similarly, compared to the decoding watermark baseline [13], our proposed method minimizes the impact on semantic generation while maintaining the same watermark strength.

In terms of watermark strength and token count, our watermark strength mainly depends on the number of watermark words in the text. The more watermark words (e.g., adjectives) included, the stronger the watermark will be.

4.3 Word Distribution and Text Fluency in Watermarked Texts

To demonstrate the susceptibility of He et al.'s [12] approach to frequency analysis, we compare the word distributions of the watermarked and clean data, as shown in Fig. 2a. We calculate the word distributions for the watermark (P_w) and benign (P_b) corpora, respectively. We then create a suspicious word set (S) by combining the top 200 adjectives from both corpora. The frequency ratio



(a) Word Frequency Changes

(b) Perplexity Distributions

Fig. 2. Effects of Watermarking on Word Frequency Ratio Changes (Top 100 Words) between Benign and Watermarked Corpora with Different Watermark Types (Top: watermarks from [12] with **red** words as synonym pairs. Bottom: COSYWA watermarks.) and Perplexity Distributions of Original and Watermarked Texts (Average Perplexity Represented by Dashed Line). (Color figure online)

$(P_b(w)/P_w(w))$ of each word in S between the benign and watermark corpora is computed.

Figure 2a depicts the noticeable disparity in adjective distribution, making the watermarking rule designed by [12] easily identifiable. In contrast, our method endeavors to conceal the watermarking rule, resulting in more covert protection.

Perplexity (PPL) serves as a pivotal metric in evaluating language model performance. It is defined as the exponential of the negative average log-likelihood attributed to a specific text under a given language model. A decrease in PPL values suggests an increase in the confidence of the language model. Language models trained on an extensive text corpus, become proficient in recognizing and learning common linguistic patterns and structures. Consequently, PPL can function as a measure of how well a text aligns with conventional characteristics. We utilized data from 200 items in the C4 dataset [22] to analyze the impact of our watermarking technique on perplexity. We employed the English GPT-2 [21] model to compute the perplexity distribution of both the original and watermarked generated text, thus enabling us to scrutinize the alterations induced by the watermarking process. Figure 2b illustrates that watermarked text exhibits elevated PPL values compared to unwatermarked text. This suggests that while COSYWA's method preserves semantic consistency, it may inadvertently compromise the fluency of the text. This phenomenon could be a result of our watermarking process promoting lexical diversity and inadvertently boosting the utilization of less common expressions. Such an impact may potentially hinder the performance of our method in open-ended answer-generation tasks.

Table 2. Performance of different watermarking approaches on CNN/DM. BS means BERTScore. RL means Rouge-L.

	Z-score	RL	BS
noun	5.38	11.54	78.01
verb	5.59	12.54	78.12
adverb	−1.64	13.18	78.28
adjective	4.82	12.92	78.31

4.4 Ablation Study

We conducted an ablation experiment to evaluate the impact of different parts of speech as watermark words on watermarking performance. We evaluated nouns, verbs, adverbs, and adjectives in the summarization task and present the results in Table 2.

Nouns and verbs as watermark words resulted in robust and detectable watermarks but compromised lexical and semantic integrity. While adverbs preserved text quality, they failed to embed detectable watermarks. Due to their prevalence in the text, nouns, and verbs resulted in a higher quantity of watermark words and greater watermark intensity. However, the increased word replacements in these cases led to a decline in semantic and lexical performance. In contrast, selecting adjectives as watermark words enabled the effective embedding of detectable watermarks while maintaining text quality. Thus, we conclude that adjectives are the most suitable part of speech for watermark words among the considered options.

4.5 Case Study

We conducted a case study on a specific instance listed in Table 3 to illustrate the enhanced semantic preservation of COSYWA. COSYWA offers improved confidentiality compared to the synonym-based method utilizing flexible replacement rules, thereby increasing the difficulty for attackers to statistically infer the watermark rules. In contrast, decoding-based watermarks [13], which modify the information content and introduce content repetition, do not preserve the semantics of the original model output as effectively as COSYWA.

Table 3. We compare COSYWA with the baseline methods and non-watermarking output. **Green** indicates the green list words, while **red** represents the red list words. **Bold** indicates the repetitive texts.

No Watermark	\n aruba audio training aruba audio training is a **good** way to get a **good** job in the field of computer based training. so get your class right from the beginning and complete the aruba audio training in short period of time. if you are a student then you should take advantage of the aruba audio training because it is the best way to get a **good** job. aruba audio training is not only a **good** way to get a **good** job but also a **good** way to learn the technical material. \n...
Ours	\n aruba audio training aruba audio training is a **superb** way to get a **full** job in the field of computer based training. so get your class right from the beginning and complete the aruba audio training in short period of time. if you are a student then you should take advantage of the aruba audio training because it is the best way to get a **desirable** job. aruba audio training is not only a **smart** way to get a **superb** job but also a **beneficial** way to learn the technical material. \n..
Synoym-based watermark [12]	\n aruba audio training aruba audio training is a **estimable** way to get a **estimable** job in the field of computer based training. so get your class right from the beginning and complete the aruba audio training in short period of time. if you are a student then you should take advantage of the aruba audio training because it is the best way to get a **estimable** job. aruba audio training is not only a **estimable** way to get a **estimable** job but also a **estimable** way to learn the technical material. \n..
Hard watermark [13]	chartered services include basic consultation services. our \ n trainees can ask for an assessment from the experts and ask for aruma auditing services. **the company offers aruma auditing services at low rates with aruma auditing services.** the company offers aruma auditing services at low rates with aruma auditing services the company offers aruma auditing services at low rates with aruma auditing services. the company offers aruma auditing services at low rates with aruma auditing services. \n...
Soft watermark [13]	\n aruba audio test questions aruba audio test questions are one of the top selling books in the market, specially aruba audio test questions. **the book offers tips and tools to help students get bigger scores in aruba audio test questions.** the book offers tips and tools to help students get bigger scores in aruba audio test questions. this book offers tips and tools to help students get bigger scores in aruba audio test questions. \n

5 Conclusion

We introduce a **CO**ntextual **SY**nonym **WA**termark (COSYWA). Our approach builds synonym spaces with an MLM and identifies acceptable and prohibited words using red and green lists from the vocabulary. We conducted experiments on three NLP tasks and validated the effectiveness of our method. We believe that our approach can be a valuable tool for detecting abusive behavior in natural language generation.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 62006138), the Key Support Project of NSFC-Liaoning Joint Foundation (No. U1908216), and the Major Scientific Research Project of the State Language Commission in the 13th Five-Year Plan (No. WT135-38). We thank all anonymous reviewers for their valuable suggestions on this work.

References

1. Adi, Y., et al.: Turning your weakness into a strength: watermarking deep neural networks by backdooring. In: USENIX-Security, pp. 1615–1631 (2018)

2. Baki, S., et al.: Scaling and effectiveness of email masquerade attacks: exploiting natural language generation. In: ACM ASIACCS, pp. 469–482 (2017)
3. Bender, E., et al.: On the dangers of stochastic parrots: can language models be too big? In: ACM FAccT, pp. 610–623 (2021)
4. Bojar, O., et al.: Findings of the 2014 workshop on statistical machine translation. In: WMT, pp. 12–58 (2014)
5. Chang, C., Clark, S.: Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. Comput. Linguist. **40**(2), 403–448 (2014)
6. Ching-Yun, C., Stephen, C.: Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In: EMNLP, pp. 1194–1203 (2010)
7. Crothers, E., et al.: Machine generated text: a comprehensive survey of threat models and detection methods. arXiv preprint arXiv:2210.07321 (2022)
8. Dehouche, N.: Plagiarism in the age of massive generative pre-trained transformers (GPT-3). ESEP **21**, 17–23 (2021)
9. Devlin, J., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
10. Giaretta, A., Dragoni, N.: Community targeted phishing: a middle ground between massive and spear phishing through natural language generation. In: Ciancarini, P., Mazzara, M., Messina, A., Sillitti, A., Succi, G. (eds.) SEDA 2018. AISC, vol. 925, pp. 86–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-14687-0_8
11. He, X., et al.: Cater: intellectual property protection on text generation APIs via conditional watermarks. In: NIPS (2022)
12. He, X., et al.: Protecting intellectual property of language generation APIs with lexical watermark. In: AAAI, vol. 36, pp. 10758–10766 (2022)
13. Kirchenbauer, J., et al.: A watermark for large language models. arXiv preprint arXiv:2301.10226 (2023)
14. Kurenkov, A.: Lessons from the GPT-4Chan controversy. The Gradient (2022)
15. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ACL (2020)
16. Li, L., et al.: Bert-attack: adversarial attack against BERT using BERT. In: EMNLP (2020)
17. Lin, C.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
18. Ng, N., et al.: Facebook FAIR's WMT19 news translation task submission. In: WMT (2020)
19. OpenAI: Chatgpt: Optimizing language models for dialogue (2022)
20. Papineni, K., et al.: Bleu: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
21. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
22. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR **21**(1), 5485–5551 (2020)
23. See, A., et al.: Get to the point: summarization with pointer-generator networks. In: ACL, pp. 1073–1083 (2017)
24. Shu, K., et al.: Mining disinformation and fake news: concepts, methods, and recent advancements. Disinformation, misinformation, and fake news in social media: emerging research challenges and opportunities, pp. 1–19 (2020)
25. Stiff, H., Johansson, F.: Detecting computer-generated disinformation. Int. J. Data Sci. Anal. **13**(4), 363–383 (2022)

26. Stribling, J., et al.: SCIgen-an automatic CS paper generator (2005)
27. Szyller, S., et al.: Dawn: dynamic adversarial watermarking of neural networks. In: ACMM, pp. 4417–4425 (2021)
28. Topkara, U., et al.: The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In: MM&Sec, pp. 164–174 (2006)
29. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: EMNLP, pp. 38–45 (2020)
30. Zhang, T., et al.: Bertscore: evaluating text generation with BERT. In: ICLR (2020)
31. Zhou, W., et al.: Bert-based lexical substitution. In: ACL, pp. 3368–3373 (2019)

Oral: Summarization and Generation

Multi-step Review Generation Based on Masked Language Model for Cross-Domain Aspect-Based Sentiment Analysis

Lei Ju[1], Xiuwei Lv[1], Zhiqiang Wang[1](✉), and Zhangwang Miao[2]

[1] Beijing Electronic Science and Technology Institute, Beijing, China
{jl,wangzq}@besti.edu.cn, 20212821@mail.besti.edu.cn
[2] State Information Center, Beijing, China

Abstract. It has been proved that supervised learning methods are effective for Aspect-Based Sentiment Analysis (ABSA). However, the supervised learning methods rely heavily on fine-grained labeled data, and many domains lack labeled data, hindering the ABSA task's effectiveness. To solve this problem, many unsupervised domain adaptation methods are proposed to transfer knowledge from the source domain with plenty of labeled data to any target domain lacking labeled data. Inspired by the implementation principle of the masked language model and the experience of predecessors, we propose a new domain adaptation paradigm named Multi-Step Review Generation (MSRG) in this paper. MSRG aims to achieve domain adaptation by generating target-domain reviews with fine-grained labels through multi-step prediction based on the source-domain labeled reviews. Pre-trained masked language model plays a significant role in MSRG. Besides, we also propose two new training strategies to adapt to MSRG. Experiment results on four benchmarks demonstrate that MSRG can achieve significant improvement compared with the state-of-the-art domain adaptation methods in both Cross-Domain End2End ABSA and Cross-Domain Aspect Extraction tasks.

Keywords: Cross Domain · Aspect-Based Sentiment Analysis · Masked Language Model

1 Introduction

Recently, Aspect-Based Sentiment Analysis (ABSA) received more and more attention [1]. With the development of deep learning, many neural network models with supervised learning methods have been proposed and have achieved surprising results for several ABSA tasks, e.g., End-to-End ABSA [2] and Aspect Extraction (AE) [3].

These supervised methods have significantly progressed in domains such as *Restaurant* and *Laptop* [4]. However, there is a problem with these methods:

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 723–735, 2023.
https://doi.org/10.1007/978-3-031-44693-1_56

supervised learning methods for ABSA need a large number of data with fine-grained labels, yet these labeled data are scarce in many new domains, which leads to these methods being given a big discount in those new domains.

Many unsupervised Domain Adaptation (DA) methods are proposed to solve this problem. These methods aim to learn domain-shared knowledge from the source domain with fine-grained labeled data and apply the knowledge to the target domain only with unlabeled data. These methods can be divided into two kinds: one focuses on learning domain-invariant representation based on feature cross domains [5], and the other aims to re-wight instances in the source domain and use it in the target domain [6].

For domain adaptation of ABSA, there are only a few studies [7–9] because of the challenges in fine-grained adaptation. However, these studies still follow the approaches of the two kinds and exit common limitations, e.g., the supervision-related knowledge for the main task solely comes from the fine-grained labeled data in the source domain, even though they can precisely reduce the discrepancy between domains. Besides, UDA [10] aims to combine the feature-based and the instance-based approaches, but UDA still exits the limitations above.

Recently, there has been a novel approach named CDRG [11] aims to generate the target-domain reviews with fine-grained labels based on the Masked Language Model (MLM; [12]) whereas the labeled source domain, then apply these generated reviews to ABSA tasks by supervised learning methods. Although CDRG obtains SOTA performance in domain adaptation for ABSA, CDRG still exits a problem in it: the process of generating the target-domain reviews is simultaneous and disposable, which leads to the relation between the generated words in the generated reviews are independent and isolate, which lead to the generated sentences cannot represent the dependency of the target domain well. This problem limits the effectiveness of CDRG in domain adaptation.

To address this problem, we propose a new approach named Multi-Step Review Generation (MSRG). Unlike CDRG, MSRG generates target-domain sentences by multi-step prediction, which can eliminate the isolation to make more correlative and dependent between tokens in generated sentences. Besides, we also propose two training strategies to adapt to MSRG. Then we further design and conduct a series of experiments to verify the correctness of MSRG. Experiment results on four benchmarks demonstrate the significant effect of MSRG compared with other domain adaptation methods [8,9].

The main contributions of this paper can be summarized as follows:

- We propose a new domain adaptation method, Multi-Step Review Generation (MSRG), based on MLM for the Cross-Domain ABSA tasks.
- To adapt to our MSRG approach, we further propose two training strategies for Cross-Domain ABSA tasks.
- Compared with other domain adaptation methods on the four benchmarks, MSRG obtains an absolute average improvement of 0.75% for Cross-Domain End2End-ABSA and 0.96% for Cross-Domain AE on the Micro-F1 score.

2 Related Work

In the literature, aspect-level sentiment classification [13], and aspect extraction [3] have been extensively studied, in which many supervised learning methods have been proposed [2]. Although gaining promising results, these methods still exit the problem that fine-grained labeled data is scarce in many domains. To solve this problem, unsupervised domain adaptation methods are proposed.

Most existing domain adaptation methods aim to learn domain-invariant representations by focusing on coarse-grained sentiment classification, including pivot-based methods [14], auto-encoders [15] and domain adversarial networks [16]. Besides, another work mainly aims to find samples helpful to the target domain from the source domain by re-weighting the instances [6]. Only a few studies have aimed to achieve cross-domain ABSA [7–10] because of the challenges in fine-grained adaptation. However, these approaches still follow the two traditional domain adaptation paradigms: feature-based and instance-based paradigms; therefore, they still exit those limitations and problems introduced in Sect. 1.

Different from the methods above, reference [11] proposed a novel method called CDRG aiming to achieve domain adaptation by generating target-domain reviews from the source domain, which gains SOTA results on the benchmarks. However, CDRG still exists some problems, e.g., the relation between the generated tokens in the sentences generated by CDRG is independent, which leads to the relation between tokens being isolated, and generated sentences cannot represent the target-domain characters and the target-domain dependency well. These problems significantly limit the effectiveness of CDRG in domain adaptation. To address these problems, we propose a domain adaptation approach named Multi-Step Review Generation (MSRG) for Cross-Domain ABSA.

3 Problem Statement

The basic problem contains two ABSA tasks, i.e., End-to-End ABSA (End2End ABSA) task and Aspect Extraction (AE) task. Following [9], we model both basic tasks as sequence labeling problems. Given a sequence of tokens $x = \{x_1, x_2, \ldots, x_n\}$ as the input, and a sequence of labels $y = \{y_1, y_2, \ldots, y_n\}$ is the output of the tasks. In the End2End ABSA task, $y_i \in$ {O, B-NEU, I-NEU, B-POS, I-POS, B-NEG, I-NEG}; and in the AE task, $y_i \in$ {B, I, O}.

Cross-Domain ABSA is based on the two ABSA tasks above. This paper focuses on unsupervised domain adaptation for Cross-Domain ABSA tasks. Formally, given a source domain contains reviews with fine-grained labels: $D^S = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$, and a target domain only contains unlabeled sentences: $D^U = \{x_i^u\}_{i=1}^{N^u}$, the goal of Cross-Domain ABSA is to predict the token labels for test sentences from the target domain: $y_i^T = f_t(x_i^t), D^T = \{x_i^t\}_{i=1}^{N^t}$.

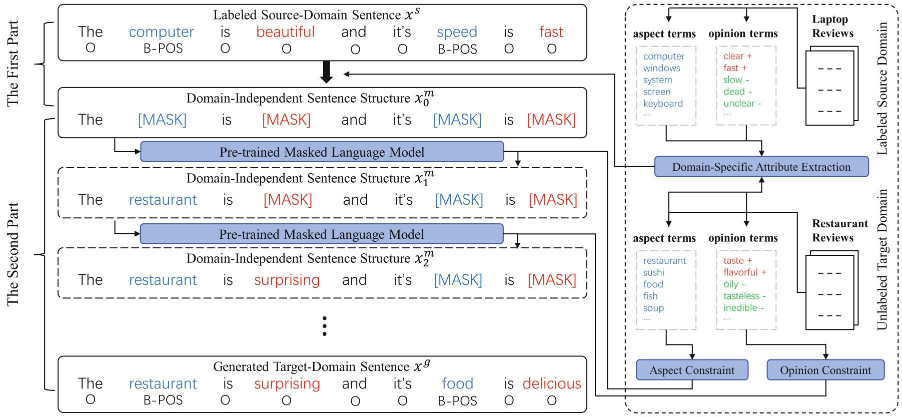

The First Part
The Second Part
Labeled Source-Domain Sentence x^s
The computer is beautiful and it's speed is fast
O B-POS O O O O B-POS O O
Domain-Independent Sentence Structure x_0^m
The [MASK] is [MASK] and it's [MASK] is [MASK]
Pre-trained Masked Language Model
Domain-Independent Sentence Structure x_1^m
The restaurant is [MASK] and it's [MASK] is [MASK]
Pre-trained Masked Language Model
Domain-Independent Sentence Structure x_2^m
The restaurant is surprising and it's [MASK] is [MASK]
Generated Target-Domain Sentence x^g
The restaurant is surprising and it's food is delicious
O B-POS O O O O B-POS O O
aspect terms
opinion terms
Laptop Reviews
computer windows system screen keyboard …
clear + fast + slow - dead - unclear - …
Labeled Source Domain
Domain-Specific Attribute Extraction
aspect terms
opinion terms
Restaurant Reviews
restaurant sushi food fish soup …
taste + flavorful + oily - tasteless - inedible - …
Unlabeled Target Domain
Aspect Constraint
Opinion Constraint

Fig. 1. The overview of the Multi-Step Review Generation approach.

4 Approach

Overview: Inspiration comes from the principle of the Masked Language Model and the experience of GPT-2 [17]. In this paper, we propose a new domain adaptation paradigm named multi-step review generation (MSRG) to solve the limitations in previous work. MSRG aims to generate target-domain reviews with fine-grained labels in multi-steps based on the pre-trained masked language model from the source-domain labeled reviews. To achieve this goal, we propose a brief yet effective two-part approach to implement MSRG, as shown in Fig. 1.

4.1 The First Part: Extract the Domain-Independent Sentence Structures

To extract the domain-independent sentence structures from the source domain, in this paper, we apply Domain Generalization [11] to achieve it. Following [11], we first extract the domain-specific attributes from both the source and target domain. The domain-specific attributes in the sentences from the source domain will be masked by the extracted domain-specific attributes above. Domain Generalization defines domain-specific attributes as words, phrases, syntactic structures, and expression styles that only occur in a specific domain. In this paper, our primary task is ABSA which mainly focuses on aspect and opinion, so we only consider aspect terms and opinion terms as domain-specific attributes.

Specifically, given labeled reviews from D^S, we can quickly obtain the aspect term list. However, due to lacking labels, we cannot directly obtain the aspect terms in the target domain and opinion terms in the target and source domain. Domain Generation applies Double Propagation [18] to obtain their list.

By Domain Generation, we can obtain both aspect and opinion term lists of the two domains. Then we remove all the attributes that occur in both domains, and we can obtain the source-domain specific aspect term lists A^S and opinion term lists O^S, and the target-domain specific aspect term lists A^T and opinion term lists O^T.

The last step of the first part is to generate the domain-independent sentence structures. Specifically, given a sentence $x^s = \{x_1, x_2, \ldots, x_n\}$ from the source domain, we only need to find each sub-sequence that is a source-domain specific attribute in either A^S or O^S and substitute each word of it with a unique token [MASK]. We will obtain the domain-independent sentence structure x^m.

4.2 The Second Part: Multi-step Prediction to Generate the Target-Domain Reviews

Regarding generating the target-domain reviews, reference [11] defined it as a text-infilling problem and proposed a viable method called Domain Specification. Specifically, for each domain-independent sentence structure $x^m = \{x_{m_1}, x_{m_2}, \ldots, x_{m_n}\}$(the number $(1, 2, \ldots, n)$ means the index of tokens in the masked sentence structures), Domain Specification pre-trains a Transformer network called BERT [12] with MLM on the unlabeled target-domain data firstly, and uses it to predict and infill each masked token in x^m in the space of target-domain specific word, and finally generates all the target-domain reviews x^g.

For the consecutive masked tokens $m_{j:j+l}$ whose length is $l+1$ between the unmasked tokens in x^m and it belongs to the aspect terms known from the labels in the source domain, the word selection of $m_{j:j+l}$ can be thus computed as follows:

$$p(x_{m_j}|x^m) = \text{TD-MLM}(x_{m_j}|x^m); \tag{1}$$

and

$$p(x_{m_{j:j+l}}|x^m) = \prod_{i=j}^{j+l-1} p(x_{m_i}|x^m); \tag{2}$$

and

$$A_{m_{j:j+l}} = \underset{x_{m_{j:j+l}}}{\arg\max}\, p(x_{m_{j:j+l}}|x^m), \tag{3}$$

where TD-MLM means the BERT model trained by MLM on the unlabeled data from the target domain D_U, and x_{m_j} means one word belongs to the vocab of the TD-MLM and is possible to appear on the index of the masked token m_j. $x_{m_{j:j+l}} \in A^T_{l+1}$ and A^T_{l+1} refers all the aspect terms which length is $l+1$ in A^T.

For masked opinion terms, Domain Specification first extracts the specific sentiment term lists (i.e., Neutral, Positive, and Negative) in both the target and source domains by Double Propagation. It uses source-domain sentiment term lists to create label sequences for each opinion term in each x^m to keep the sentiment alignment, then predicts all the masked opinion tokens by the method similar to predicting the masked aspect tokens for each x_m.

Domain Specification can obtain the generated target-domain reviews x^g by the methods above. However, a problem exists on x^g generated by Domain Specification. The problem is that the relationship between the tokens infilled by Domain Specification is independent. The reason for independence is that the infill of all the masked tokens in each x_m is disposable and simultaneous.

Let us suppose, for example, given a masked domain-independent sentence structure $x^m = \{token_1, [\mathrm{MASK}]_2, token_3, [\mathrm{MASK}]_4, token_5\}$, now predicting the tokens in the index of [MASK]$_2$ and [MASK]$_4$ and the result is that substitute [MASK]$_2$ with the $token_2$ and substitute [MASK]$_4$ with the $token_4$, and the result review $x^g = \{token_1, token_2, token_3, token_4, token_5\}$. The word selection of the two masked tokens is as follows:

$$token_2 = \arg\max_{x_{m_2}} p(x_{m_2}|x^m); \tag{4}$$

and

$$token_4 = \arg\max_{x_{m_4}} p(x_{m_4}|x^m), \tag{5}$$

where $x^m = \{token_1, [\mathrm{MASK}]_2, token_3, [\mathrm{MASK}]_4, token_5\}$.

According to the equation above, we can learn that the word selection of [MASK]$_2$ and [MASK]$_4$ is independent. Simply, the word selection of [MASK]$_2$ is the conditional probability in the condition of x^m, so it is independent of the word selection of [MASK]$_4$, vice versa, which cause the isolation between the generated tokens. To solve this problem and improve the dependency between all the tokens in the generated target-domain sentences, in this part, we propose a new method called Multi-Step Prediction (MSP).

Specifically, given a masked sentence structure $x^m = \{x_{m_1}, x_{m_2}, \ldots, x_{m_n}\}$, according to MLM, the state of x^m will be used to predict the original token with cross-entropy loss. In other words, the prediction of each masked token x_{m_j} in x^m will be produced as a conditional probability $p(x_{m_j}|x^m)$ by MLM, and the modeling of cross entropy loss of MLM is based on the conditional probabilities. Actually, the language model is modeling the conditional probability of sequence, and the specific presentation of the formula about the conditional probabilities above is as follows:

$$p(x_{m_j}|x^m) = p(x_{m_j}|x_{m_1}, \ldots, x_{m_{j-1}}, x_{m_{j+1}}, \ldots, x_{m_n}) \tag{6}$$

where $x^m = \{x_{m_1}, x_{m_2}, \ldots, x_{m_n}\}$.

According to GPT-2 [17], given a sequence of symbols $(s_1, s_2, \ldots, s_n)$, modeling of GPT-2 is based on the joint probabilities $p(s_{n-k}, \ldots, s_n|s_1, \ldots, s_{n-k-1})$, and the joint probabilities can be factorized and estimated over symbols as the product of conditional probabilities [19,20]:

$$p(x) = \prod_{i=1}^{n} p(s_n|s_1, s_2, \ldots, s_{n-1}) \tag{7}$$

Inspiration comes from the factorizing joint probabilities and MLM. We propose a multi-step prediction approach to predict the masked token in the domain-independent sentence structures and infill it. Specifically, given a masked sentence structure $x_0^m = \{token_1, [\text{MASK}]_2, token_3, [\text{MASK}]_4, \ldots, token_n\}$, the first step is to predict the masked token $[\text{MASK}]_2$ based on MSP as follows:

$$token_2 = \arg\max_{x_{m_2}} p(x_{m_2}|x_0^m) \tag{8}$$

By the first step of prediction based on the pre-trained masked language model, we chose the $token_2$ to substitute the masked token $[\text{MASK}]_2$, and we can obtain the sentence structure $x_1^m = \{token_1, token_2, token_3, [\text{MASK}]_4, \ldots, token_n\}$. One of the masked tokens in x_0^m is infilled and we obtain x_1^m. Then, we continue to predict the next masked token $[\text{MASK}]_4$ in x_1^m:

$$token_4 = \arg\max_{x_{m_4}} p(x_{m_4}|x_1^m) \tag{9}$$

Now, we can substitute $[\text{MASK}]_4$ with $token_4$, and we can obtain the sentence structure $x_2^m = \{token_1, token_2, token_3, token_4, \ldots, token_n\}$. Moreover, we can find that the prediction of $token_4$ is in the condition of x_1^m, and it shows that the $token_4$ is correlative with the $token_2$. Then we predict all the remaining masked tokens in x_2^m one by one until all the masked tokens are predicted and substituted. Now, we wholly generate a complete sentence x^g without the masked tokens [MASK]:

$$x^g = \{token_1, token_2, token_3, token_4, \ldots, token_n\} \tag{10}$$

According to (8), (9) and (10), we can find the differency between x^g generated by MSP and x^g generated by Domain Specification. Compared with Domain Specification, the tokens in x^g generated by MSP are full of correlation, and the tokens generated by Domain Specification are isolation, which creates more dependency between the words in sentences generated by MSP.

We can easily obtain each generated target-domain review x^g and its sequence labels from x^s through the first part and the second part. Finally, we can obtain all the generated target-domain reviews $D^G = \{(x_i^g, y_i^s)\}_{i=1}^{N^s}$.

4.3 Training Strategies

After obtaining D^G, two training strategies for ABSA tasks are proposed in [11]: one is Independent Training: training the BERT model only on D^G, and another is Merge Training: the BERT model is training on the merge corpus $D^S \cup D^G$. As we all know, the generation of tokens by CDRG is simultaneous, yet the generation of tokens by MSRG is one by one and has an order. To adapt to MSRG and enhance the robustness of models, we further propose two new training strategies: Bi-directional Generation Independent Training (BiIndep Training) and Bi-directional Generation Merge Training (BiMerge Training). Specifically, for each domain-independent sentence structure x^m from the source domain, we infill the masked tokens in x^m from front to back of the sentence by MSP first. We can obtain the generated target-domain sentence x_1^g with labels, then we apply MSP to infill the masked tokens in x^m from back to front, and we can gain a generated target-domain sentence x_2^g with labels. Now, all the x_1^g form the generated target-domain reviews set D_1^G and all the x_2^g form the set D_2^G. Finally, for BiIndep Training, the model is training on $D_1^G \cup D_2^G$, and for BiMerge Training, the model is training on the corpus $D^S \cup D_1^G \cup D_2^G$. Besides, for Independent Training and Merge Training to test the MSRG, the D^G only contains D_1^G.

5 Experiment

5.1 Datasets

Table 1. Statistics of the four benchmark datasets

Dataset	Domain	Sentences	Training	Testing
L	Laptop	3845	3045	800
R	Restaurant	6035	3877	2158
D	Device	3836	2557	1279
S	Service	2239	1492	747

Experiments are conducted on four benchmark datasets: Laptop (L), Restaurant (R), Device (D), and Service (S). L means the reviews from the laptop domain [4], and R contains all the restaurant datasets from SemEval 2014 [4], 2015 [21] and 2016 [22] ABSA challenge. D is all the digital device reviews from [23], and S contains some reviews about web services introduced by [24]. Detailed statistics are shown in Table 1.

5.2 Experiment Settings and Implementation Details

To compare with the previous work fairly, we conduct experiments the same as [11]. The experiments are on ten transfer pairs with the domains from the four benchmark datasets. The training data contains the labeled training data from the source domain and the unlabeled training data from the target domain. In addition, the testing data from the target domain is used for evaluation experiments, and we use the Micro-F1 score as our evaluation metric for comparing fairly. The hyperparameters are kept pace with [11]. In the stage of re-training the BERT-based model with MLM, the learning rate and the batch size are set to 3e-5 and 32. Moreover, in the stage of two training strategies for two ABSA tasks, the learning rate, batch size, and dropout rate are set to 5e-5, 32, and 0.1. Furthermore, we employ Adam optimizer [25]. Besides, we ran each experiment 5 times and reported the average of 5 runs as the results for all the experiments.

5.3 Baselines

To show the effectiveness of our MSRG-based approach, we compare MSRG with several baselines in Domain Adaptation as follows:

- DP [18]: the Double Propagation method introduced in Sect. 4.1.
- Hier-Joint [7]: A syntactic rule-based auxiliary tasks based on RNN.
- RNSCN [8]: A recursive neural structural correspondence network that combines syntactic structures.
- AD-SAL [9]: A DA method based on Selective Adversarial Learning.
- $\mathrm{BERT_B}$ and $\mathrm{BERT_E}$: $\mathrm{BERT_B}$ is the BERT-based uncased model from [12], and the $\mathrm{BERT_E}$ is another $\mathrm{BERT_{base}}$ uncased model from [26] and it is pre-trained on E-commerce reviews.
- $\mathrm{BERT_B}$-UDA and $\mathrm{BERT_E}$-UDA [10]: A DA method based on $\mathrm{BERT_B}$ and $\mathrm{BERT_E}$ and that unifies feature and instance.
- $\mathrm{BERT_B}$-CDRG-X and $\mathrm{BERT_E}$-CDRG-X [11]: The CDRG-based approaches based on $\mathrm{BERT_B}$ and $\mathrm{BERT_E}$. The X refers the training strategies.
- $\mathrm{BERT_B}$-MSRG-X and $\mathrm{BERT_E}$-MSRG-X: The MSRG-based approaches our propose. The X also refers the training strategies.

5.4 Experiment Results and Analysis

We report the experiment results in Table 2 for cross-domain End2End-ABSA and Table 3 for cross-domain AE. As shown in Table 2, we can observe that MSRG-based approaches acquire better performance than the state-of-the-art domain adaptation methods (i.e., CDRG) on most transfer pairs for cross-domain End2End-ABSA. Besides, on the one hand, the performance in the basic of $\mathrm{BERT_E}$ is better than the performance in the basic of $\mathrm{BERT_B}$; on the other hand, the performance in the condition of Merge Training is better than the performance in the condition of Independent Training. Under Independent Training, MSRG outperforms CDRG by an absolute improvement of 0.86% based on $\mathrm{BERT_B}$ and 0.14% based on $\mathrm{BERT_E}$. Moreover, under Merge Training, MSRG outperforms CDRG by an absolute improvement of 0.62% based on $\mathrm{BERT_B}$ and 0.50% based on $\mathrm{BERT_E}$. In general, the experiment results demonstrate the effectiveness of MSRG.

As shown in Table 3, we can find that the trend of the improvement of MSRG in cross-domain AE is similar to it in cross-domain End2End-ABSA. Under Independent Training, MSRG outperforms CDRG by an absolute improvement of 1.55% based on $\mathrm{BERT_B}$ and 0.08% based on $\mathrm{BERT_E}$. Moreover, under Merge Training, MSRG outperforms CDRG by an absolute improvement of 0.87% based on $\mathrm{BERT_B}$ and 0.76% based on $\mathrm{BERT_E}$. Besides, We can observe that the improvement of MSRG in cross-domain AE is more significant than in cross-domain End2End-ABSA, which is reasonable as AE is a subtask of ABSA. In summary, MSRG can solve the problems that exist in CDRG to a certain extent.

Moreover, when we run MSRG on the two new training strategies (i.e., BiIndep Training and BiMerge Training), MSRG gains better performance. When doing the experiment in the condition of BiMerge Training and being based on $\mathrm{BERT_E}$, MSRG obtains the best performance both for cross-domain End2End-ABSA and cross-domain AE. Compared with CDRG, MSRG outperforms CDRG by an absolute improvement of 0.75% for cross-domain End2End-ABSA and 0.96% for cross-domain AE, demonstrating the effectiveness of our two new training strategies.

Table 2. Experiment results of different methods for Cross-Domain End2End ABSA based on Micro-F1.

Methods	S→R	L→R	D→R	R→S	L→S	D→S	R→L	S→L	R→D	S→D	Average
DP	34.47	34.47	34.47	18.31	18.31	18.31	16.63	16.63	19.03	19.03	22.97
Hier-Joint	31.10	33.54	32.87	15.56	13.90	19.04	20.72	22.65	24.53	23.24	23.72
RNSCN	33.21	35.65	34.60	20.04	16.59	20.03	26.63	18.87	33.26	22.00	26.09
AD-SAL	41.03	43.04	41.01	28.01	27.20	26.62	34.13	27.04	35.44	33.56	33.71
$\mathrm{BERT_B}$	44.66	40.38	40.32	19.48	25.78	30.31	31.44	30.47	27.55	33.96	32.44
$\mathrm{BERT_B}$-UDA	47.09	45.46	42.68	33.12	27.89	28.03	33.68	34.77	34.93	32.10	35.98
$\mathrm{BERT_B}$-CDRG-Indep Training	44.46	44.96	39.42	34.10	33.97	31.08	33.59	26.81	25.25	29.06	34.27
$\mathrm{BERT_B}$-CDRG-Merge Training	47.92	49.79	47.64	35.14	38.14	37.22	38.68	33.69	27.46	34.08	38.98
$\mathrm{BERT_B}$-MSRG-Indep Training	45.56	49.35	39.56	35.80	32.80	28.92	32.05	31.75	24.89	30.58	35.13
$\mathrm{BERT_B}$-MSRG-Merge Training	48.99	53.71	49.18	38.90	38.74	36.99	37.16	32.09	25.03	35.22	39.60
$\mathrm{BERT_B}$-MSRG-BiIndep Training	45.97	49.81	39.98	36.21	33.26	29.34	32.45	32.21	25.11	31.22	35.56
$\mathrm{BERT_B}$-MSRG-BiMerge Training	49.31	54.01	49.48	39.22	38.92	37.32	37.60	32.38	25.28	35.59	39.91
$\mathrm{BERT_E}$	51.34	45.40	42.62	24.44	23.28	28.18	39.72	35.04	33.22	33.22	35.65
$\mathrm{BERT_E}$-UDA	53.97	49.52	51.84	30.67	27.78	34.41	43.95	35.76	**40.35**	38.05	40.63
$\mathrm{BERT_E}$-CDRG-Indep Training	51.01	54.56	54.33	**42.52**	39.28	36.98	40.23	33.41	30.56	32.05	41.49
$\mathrm{BERT_E}$-CDRG-Merge Training	53.09	57.96	54.39	40.85	**42.96**	38.83	**45.66**	35.06	31.62	34.22	43.46
$\mathrm{BERT_E}$-MSRG-Indep Training	53.54	56.62	50.86	36.72	41.30	38.58	41.50	34.49	28.39	34.31	41.63
$\mathrm{BERT_E}$-MSRG-Merge Training	54.41	58.09	55.41	39.13	41.88	42.94	42.92	36.35	30.49	37.96	43.96
$\mathrm{BERT_E}$-MSRG-BiIndep Training	53.59	56.64	50.94	36.75	41.34	38.60	41.56	34.51	28.42	34.35	41.67
$\mathrm{BERT_E}$-MSRG-BiMerge Training	**54.72**	**58.43**	**55.75**	39.33	42.09	**43.21**	43.20	**36.52**	30.69	**38.14**	**44.21**

5.5 Ablation Study

To investigate the effectiveness of MSRG, we further conduct the ablation study. In this part, we propose to verify the closeness between the reviews from the target-domain test set D^T and the D^G generated by different methods with Maximum Mean Discrepancy (MMD) [27]. MMD can measure the similarity between the reviews; the lower value of MMD means more similarity. The results are shown in Table 4. $\mathrm{BERT_E}$ means only using $\mathrm{BERT_E}$ without extra re-training. We can observe that the MMD between D^T and D^G generated by MSRG is the smallest, which indicates that their distance is smaller than other methods and they are more similar. Generally, MSRG can generate reviews whose distribution is closer to the target-domain reviews than other methods.

Table 3. Experiment results of different methods for Cross-Domain Aspect Extraction (AE) based on Micro-F1.

Methods	S→R	L→R	D→R	R→S	L→S	D→S	R→L	S→L	R→D	S→D	Average
DP	37.63	37.63	37.63	19.74	19.74	19.74	19.79	19.79	21.82	21.82	25.53
Hier-Joint	46.39	48.61	42.96	27.18	25.22	29.28	34.11	33.02	34.81	35.00	35.66
RNSCN	48.89	52.19	50.39	30.41	31.21	35.50	47.23	34.03	**46.16**	32.41	40.84
AD-SAL	52.05	56.12	51.55	39.02	38.26	36.11	45.01	35.99	43.76	**41.21**	43.91
$\mathrm{BERT_B}$	54.29	46.74	44.63	22.31	30.66	33.33	37.02	36.88	32.03	38.06	37.60
$\mathrm{BERT_B}$-UDA	56.08	51.91	50.54	34.62	32.49	34.52	46.87	43.98	40.34	38.36	42.97
$\mathrm{BERT_B}$-CDRG-Indep Training	53.79	55.13	50.07	41.74	44.14	37.10	40.18	33.22	30.78	34.97	42.11
$\mathrm{BERT_B}$-CDRG-Merge Training	56.26	60.03	52.71	42.36	47.08	41.85	46.65	39.51	32.60	36.97	45.60
$\mathbf{BERT_B}$-MSRG-Indep Training	54.65	61.00	51.71	43.46	42.36	38.81	42.41	37.92	30.06	34.23	43.66
$\mathbf{BERT_B}$-MSRG-Merge Training	57.42	64.43	54.09	45.95	47.31	41.12	47.98	39.08	29.65	37.69	46.47
$\mathbf{BERT_B}$-MSRG-BiIndep Training	55.52	62.18	52.59	44.15	43.22	39.47	43.29	38.52	30.42	34.85	44.42
$\mathbf{BERT_B}$-MSRG-BiMerge Training	57.90	64.97	54.57	46.35	47.79	41.56	48.39	39.40	29.91	37.99	46.88
$\mathrm{BERT_E}$	57.56	50.42	45.71	26.50	25.96	30.40	44.18	41.78	35.98	35.13	39.36
$\mathrm{BERT_E}$-UDA	59.07	55.24	56.40	34.21	30.68	38.25	54.00	**44.25**	42.40	40.83	45.53
$\mathrm{BERT_E}$-CDRG-Indep Training	58.75	65.81	59.61	**50.68**	51.25	40.17	49.17	41.61	33.34	36.97	48.74
$\mathrm{BERT_E}$-CDRG-Merge Training	59.17	68.62	58.85	47.61	**54.29**	42.20	**55.56**	41.77	35.43	36.53	50.00
$\mathbf{BERT_E}$-MSRG-Indep Training	61.49	67.06	56.57	44.66	53.48	41.28	50.20	43.72	32.68	37.04	48.82
$\mathbf{BERT_E}$-MSRG-Merge Training	63.48	68.98	59.79	45.74	52.53	45.76	53.69	43.32	34.08	40.23	50.76
$\mathbf{BERT_E}$-MSRG-BiIndep Training	61.57	67.13	56.64	44.70	53.52	41.32	50.26	43.76	32.72	37.06	48.87
$\mathbf{BERT_E}$-MSRG-BiMerge Training	**63.75**	**69.31**	**60.06**	45.90	52.71	**45.92**	53.91	43.46	34.09	40.49	**50.96**

Table 4. Maximum Mean Discrepancy Experiment Results

	Source	$\mathrm{BERT_E}$	CDRG	MSRG
S→R	0.3496	0.2411	0.1477	**0.1327**
L→R	0.3155	0.2978	0.2072	**0.1843**
D→R	0.3504	0.3178	0.1650	**0.1335**
R→S	0.3330	0.2977	0.1654	**0.1269**
L→S	0.1966	0.1945	0.1518	**0.1287**
D→S	0.1804	0.1628	0.0941	**0.0860**
R→L	0.3135	0.2563	0.1080	**0.0837**
S→L	0.2767	0.1652	0.1763	**0.1481**
R→D	0.3476	0.2954	**0.1433**	0.1569
S→D	0.1765	0.0929	0.0788	**0.0697**
Average	0.2840	0.2322	0.1438	**0.1251**

6 Conclusion and Future Work

In this paper, we propose a new domain adaptation approach named Multi-Step Review Generation (MSRG) and two new training strategies adapting to MSRG for Cross-Domain ABSA tasks to solve the problems existing in previous studies, and we also propose a brief yet effective two-part approach to implement MSRG. Compared with other domain adaptation methods, the experiment results on four benchmarks demonstrate the significant effectiveness and stable improvement of MSRG both for the Cross-Domain End2End-ABSA task and for the Cross-Domain AE task. However, MSRG only re-trains the pre-trained model BERT on relative corpus by the cross entropy of MLM. It has yet to consider applying feature-based and instance-based domain adaptation methods in the review generation process. How combining the two traditional domain adaptation methods based on review generation is a promising direction in our future work.

Acknowledgement. This paper's work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 328202203, 20230045Z0114), and China Postdoctoral Science Foundation funded project (Grant No. 2019M650606).

References

1. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press (2020)
2. Zhang, M., Zhang, Y., Vo, D.-T.: Neural networks for open domain targeted sentiment. In: EMNLP (2015)
3. Xu, H., Liu, B., Shu, L., Yu Philip, S.: Double embeddings and CNN-based sequence labeling for aspect extraction. In: ACL (2018)
4. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, A.: SemEval-2014 task 4: aspect based sentiment analysis. In: SemEval-2014 (2014)
5. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: ACL (2007)
6. Dredze, M., Kulesza, A., Crammer, K.: Multi-domain learning by confidence-weighted parameter combination. Mach. Learn. **79**(1–2), 123–149 (2010)
7. Ding, Y., Yu, J., Jiang, J.: Recurrent neural networks with auxiliary labels for crossdomain opinion target extraction. In: AAAI (2017)
8. Wang, W., Pan, S.J.: Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In: ACL (2018)
9. Li, Z., Li, X., Wei, Y., Bing, L., Zhang, Y., Yang, Q.: Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. In: EMNLP (2019)
10. Gong, C., Yu, J., Xia, R.: Unified feature and instance based domain adaptation for end-to-end aspect-based sentiment analysis. In: EMNLP (2020)
11. Yu, J., Gong, C., Xia, R.: Cross-domain review generation for aspect-based sentiment analysis. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 4767–4777 (2021)
12. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
13. Wang, S., Mazumder, S., Liu, B., Zhou, M., Chang, Y.: Target-sensitive memory networks for aspect sentiment classification. In: ACL (2018)
14. Yu, J., Jiang, J.: Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In: EMNLP (2016)
15. Zhuang, F., Cheng, X., Luo, P., Pan, S.J., He, Q.: Supervised representation learning: transfer learning with deep autoencoders. In: IJCAI (2015)
16. Li, Z., Wei, Y., Zhang, Y., Yang, Q.: Hierarchical attention transfer network for cross-domain sentiment classification. In: AAAI (2018)
17. Radford, A., Jeffrey, W., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

18. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
19. Jelinek, F., Mercer, R.L.: Interpolated estimation of Markov source parameters from sparse data. In: Proceedings of the Workshop on Pattern Recognition in Practice, Amsterdam, The Netherlands: North-Holland, May 1980
20. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
21. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 task 12: aspect based sentiment analysis. In: SemEval-2015 (2015)
22. Pontiki, M., Galanis, D., Papageorgiou, H., et al.: SemEval-2016 task 5: aspect based sentiment analysis. In: SemEval-2016 (2016)
23. Toprak, C., Jakob, N., Gurevych, I.: Sentence and expression level annotation of opinions in user-generated discourse. In: ACL (2010)
24. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: SIGKDD (2004)
25. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Xu, H., Liu, B., Shu, L., Yu Philip, S.: BERT post-training for review reading comprehension and aspect-based sentiment analysis. In: NAACL (2019)
27. Gretton, A., Borgwardt, K.M., Rasch, M.J., Scholkopf, B., Smola, A.: A kernel two-sample test. J. Mach. Learn. Res. **13**(1), 723–773 (2012)

rT5: A Retrieval-Augmented Pre-trained Model for Ancient Chinese Entity Description Generation

Mengting Hu[1], Xiaoqun Zhao[1], Jiaqi Wei[1], Jianfeng Wu[2], Xiaosu Sun[1],
Zhengdan Li[1], Yike Wu[3,4(✉)], Yufei Sun[1], and Yuzhi Zhang[1]

[1] College of Software, Nankai University, Tianjin, China
mthu@nankai.edu.cn
[2] College of Computer Science, Nankai University, Tianjin, China
[3] School of Journalism and Communication, Nankai University, Tianjin, China
wuyike@nankai.edu.cn
[4] Convergence Media Research Center, Nankai University, Tianjin, China

Abstract. Ancient Chinese, the natural language of ancient China, serves as the key to understanding and propagating Chinese rich history and civilization. However, to facilitate comprehension and education, human experts previously need to write modern language descriptions for special entities, such as persons and locations, out of ancient Chinese texts. This process requires specialized knowledge and can be time-consuming. To address these challenges, we propose a new task called Ancient Chinese Entity Description Generation (ACEDG), which aims to automatically generate modern language descriptions for ancient entities. To address ACEDG, we propose two expert-annotated datasets, XunZi and MengZi, each containing ancient Chinese texts, and some of them have been annotated with entities and their descriptions by human experts. To leverage both labeled and unlabeled texts, we propose a retrieval-augmented pre-trained model called rT5. Specifically, a pseudo-parallel corpus is constructed using retrieval techniques to augment the pre-training stage. Subsequently, the pre-trained model is fine-tuned on our high-quality human-annotated entity-description corpus. Our experimental results, evaluated using various metrics, demonstrate the effectiveness of our method. By combining retrieval techniques and pre-training, our approach significantly advances the state-of-the-art performance in the ACEDG task compared with strong pre-trained models.

Keywords: Ancient Chinese · Entity Description Generation

1 Introduction

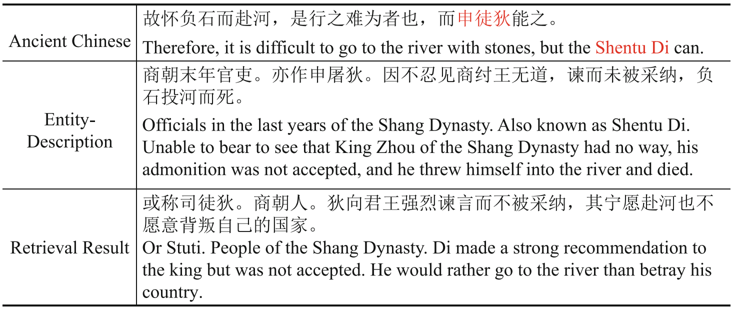
Throughout the extensive history of China, ancient Chinese texts are the essence of national thought and cultural spirit. They serve as a bridge connecting history

This research is supported by the youth program of National Science Fund of Tianjin, China (Grant No. 22JCQNJC01340), the Fundamental Research Funds for the Central University, Nankai University (Grant No. 63221028 and No. 63232114).

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 736–748, 2023.
https://doi.org/10.1007/978-3-031-44693-1_57

with modern culture and transmit valuable information to future generations. Notably, ancient Chinese differs from modern Chinese in various aspects, including vocabulary and syntax. To aid in comprehension and education, there are several natural language processing tasks focused on ancient Chinese, such as named entity recognition [1], poem generation [2], and couplet generation [3].

Different from these works, we deal with ancient Chinese from the entity description view. That is to say, human experts previously need to annotate ancient Chinese entities with modern Chinese descriptions, which is very time-consuming and relies heavily on experts' knowledge. To save human effort, we propose a new task called Ancient Chinese Entity Description Generation (ACEDG), aiming to automatically generate modern language descriptions for ancient entities. As shown in Fig. 1, "申徒狄" (Shentu Di) is a person entity in the ancient Chinese sentence. ACEDG aims to generate a modern Chinese description considering the contexts. In this example, the entity description presents a brief biography of this person.



Ancient Chinese	故怀负石而赴河，是行之难为者也，而申徒狄能之。 Therefore, it is difficult to go to the river with stones, but the Shentu Di can.
Entity-Description	商朝末年官吏。亦作申屠狄。因不忍见商纣王无道，谏而未被采纳，负石投河而死。 Officials in the last years of the Shang Dynasty. Also known as Shentu Di. Unable to bear to see that King Zhou of the Shang Dynasty had no way, his admonition was not accepted, and he threw himself into the river and died.
Retrieval Result	或称司徒狄。商朝人。狄向君王强烈谏言而不被采纳，其宁愿赴河也不愿意背叛自己的国家。 Or Stuti. People of the Shang Dynasty. Di made a strong recommendation to the king but was not accepted. He would rather go to the river than betray his country.

Fig. 1. An example of ACEDG. Our purpose is to generate modern Chinese descriptions from ancient entities. The retrieval result demonstrates that retrieved texts have the potential of providing extra knowledge to help generate.

The above example illustrates that interpreting ancient Chinese texts requires not only contextual information but also expert knowledge. However, acquiring such knowledge relies heavily on human experts with a solid background in history and literature. Additionally, accurately conveying the importance of the entity category presented in the original sentence poses another challenge. This is due to the nature of the language itself, as ancient Chinese is markedly distinct from modern Chinese in terms of sentence structure and vocabulary. Ancient Chinese sentences feature complex combinations of content and function words, alongside colloquial characters, making them more challenging to comprehend than modern Chinese.

To deal with the above challenges, we propose a retrieval-augmented pre-trained model called rT5. Concretely, we first construct pseudo-parallel corpora based on different retrieval algorithms. By comparing different evaluation metrics, the corpus generated by the best retrieval algorithm is chosen as the pre-training data for the model. Subsequently, we utilize retrieval techniques to augment the pre-training phase and fine-tune the pre-trained model on our high-quality human-annotated entity-description corpus. To verify the effectiveness of our method, we conduct experiments on two expert-annotated datasets, XunZi and MengZi. Each dataset contains ancient Chinese texts, and some of them have been annotated with entities and their descriptions by human experts. Extensive experimental results suggest the effectiveness of our method. By combining retrieval techniques and pre-training with fine-tuning, our approach significantly advances the state-of-the-art performance in the ACEDG task compared with strong pre-trained models.

In summary, the contributions of this work are three-fold:

- We propose a new task called Ancient Chinese Entity Description Generation (ACEDG) and propose two expert-annotated datasets, XunZi and MengZi. To the best of our knowledge, though there are some works dealing with ancient Chinese, we are the first to focus on ancient Chinese entity description generation.
- To leverage both labeled and unlabeled texts, we propose a retrieval-augmented pre-trained model called rT5 for promoting ACEDG task.
- Our experimental results demonstrate the effectiveness of our method. By combining retrieval techniques and pre-training, our approach significantly advances the state-of-the-art performance in the ACEDG task compared with strong pre-trained models.

Table 1. The statistics of entity-description pairs for each type in the datasets.

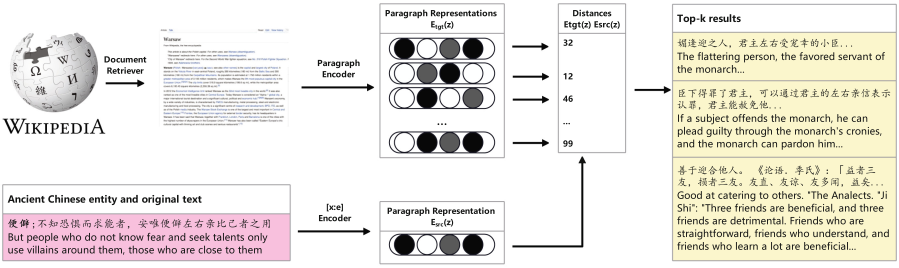
category	Person	Location	Thing	Literature	Official	Institution	Time	Knowledge
XunZi	487	114	78	689	77	9	35	3822
MengZi	141	14	52	135	4	0	0	126

2 Dataset Construction

Two kinds of corpora are constructed in the experiments, including expert-annotated datasets and retrieval-based pre-trained corpus. The first one is high-quality ancient Chinese entities and corresponding entity descriptions. Due to its high cost, the scale is relevantly small. Then the second one is constructed based on retrieval techniques, which is cost-friendly.

Expert-Annotated Datasets. To address ACEDG, we propose two expert-annotated datasets, namely XunZi and MengZi, which consist of ancient Chinese texts that have been annotated with entities and their corresponding descriptions by experts, who are professors at the school of history. The manual annotation of entities and their descriptions is conducted on the traditional Chinese Culture books of XunZi and MengZi, resulting in a total of 5311 XunZi entity-description pairs and 473 MengZi entity-description pairs. The entities are classified into eight types, including person, location, thing, literature, official, institution, time and knowledge. Table 1 presents the statistics of each entity type in the datasets.

Retrieval-Based Pre-trained Corpus. Since we only annotate a small fraction of texts in the original XunZi and MengZi books, there still left many unlabeled ancient Chinese texts. To leverage them, we first use the original XunZi book to pre-train a BERT model [4], called XunZiBERT. Then the pre-trained model is utilized to fine-tune for named entity recognition (NER), with F1 score of more than 93.5%. The NER model is further leveraged to annotate entities for two books, respectively. The pseudo-parallel corpus of ancient Chinese entity descriptions is constructed by retrieving entity information of ancient Chinese entities from more than 910,000 Wikipedia entries. Among them, the conversion of traditional Chinese characters to simplified characters is completed by Opencc. According to the principle of the retrieval enhancement model, dual encoders are used to encode the information of ancient Chinese text entities and Wikipedia respectively. The most relevant item is retrieved. The details are shown in Fig. 2.


Wikipedia
Document Retriever
Warsaw
Paragraph Encoder
Paragraph Representations $E_{tgt}(z)$
...
Distances Etgt(z) Esrc(z)
32
12
46
...
99
Top-k results
Ancient Chinese entity and original text
[x:e] Encoder
Paragraph Representation $E_{src}(z)$

便僻；不知恐惧而求能者，安唯便僻左右亲比己者之用
But people who do not know fear and seek talents only use villains around them, those who are close to them

媚逢迎之人，君主左右受宠幸的小臣...
The flattering person, the favored servant of the monarch...

臣下得罪了君主，可以通过君主的左右亲信表示认罪，君主能赦免他...
If a subject offends the monarch, he can plead guilty through the monarch's cronies, and the monarch can pardon him...

善于迎合他人。《论语．季氏》：「益者三友，损者三友。友直、友谅、友多闻，益矣...
Good at catering to others. "The Analects. "Ji Shi": "Three friends are beneficial, and three friends are detrimental. Friends who are straightforward, friends who understand, and friends who learn a lot are beneficial...

Fig. 2. An illustration of retrieval process. We first retrieve data from Wikipedia, divide it into paragraphs of approximately equal length, and encode each paragraph separately with the target encoder. Each time a pair of entity and entity descriptions is generated, the entity is encoded, and the most suitable paragraph is used as the entity description by dot multiplication.

3 Methodology

3.1 Problem Formulation and Overview

In this paper, we define a practical problem, i.e. ACEDG, and tackle this task by leveraging external knowledge from retrieval. In practice, it is difficult to describe an ancient entity only by its context. The reason is that a good description requires additional expertise and knowledge background. As the example shown in Fig. 1, retrieval result, i.e. *"Or Stuti. People of the Shang Dynasty. Di made a strong recommendation to the king but was not accepted. He would rather go to the river than betray his country."*, contains rich knowledge for improving the quality of description. Therefore, in addition to the original text context information in the ancient text, we introduce pre-training and retrieval methods. A pseudo-parallel corpus is constructed through retrieval enhancement, which is further used to pre-train the sequence-to-sequence learning model.

Formally, given a sentence $\boldsymbol{x}$ in ancient Chinese and an entity $\boldsymbol{e}$ in this sentence, ACEDG task aims to generate the entity's description $\boldsymbol{y}$ in modern Chinese. Our method enhances ACEDG with retrieved knowledge $\boldsymbol{k}$, which mainly comprises the following three stages:

- **Knowledge Retrieval:** The ancient Chinese sentence $\boldsymbol{x}$ and entity $\boldsymbol{e}$ are leveraged to retrieve modern Chinese knowledge $\boldsymbol{k}$, which is regarded as the pseudo-parallel pairs for the next pre-training stage.
- **Pseudo-Parallel Corpus Pre-Training:** ACEDG relies heavily on human experts' historical knowledge. Though pseudo-parallel pairs are noisy, they still provide rich references for ACEDG. Thus, the retrieved sequences are utilized to pre-train a generation model.
- **Knowledge Enhanced Fine-Tuning:** Finally, the pre-trained model is further fine-tuned on expert-annotated datasets, i.e. XunZi and MengZi in a knowledge-enhanced sequence-to-sequence learning manner.

Table 2. Evaluation results for various retrieval methods on the testing set of XunZi, in terms of BLEU (%), Rouge-1 (%), Rouge-2 (%), Rouge-L (%), and Meteor (%).

Methods	BLEU-1	BLEU-2	Rouge-1	Rouge-2	Rouge-L	Meteor
BM25	8.17	0.39	9.05	0.57	7.63	3.9
Cosine	6.38	0.11	6.14	0.06	7.63	3.9
IDF	6.44	0.07	6.43	0.06	5.26	2.39
Jaccard	**10.22**	0.21	13.05	0.35	9.66	5.15
mContriever ($\boldsymbol{e}$)	4.79	0.45	6.52	0.06	5.32	2.43
mContriever ($[\boldsymbol{x}; \boldsymbol{e}]$)	7.29	**1.63**	**13.26**	**2.18**	**10.27**	**9.33**

3.2 Knowledge Retrieval

To obtain external modern Chinese knowledge, we retrieve the sentences most related to ancient Chinese entities from Wikipedia's Modern Chinese corpus Z. Concretely, the retrieval model mContriever [5] is adopted. Given an input sentence $\boldsymbol{x}$ and one of its entities $\boldsymbol{e}$ in ancient Chinese, the retrieval model first selects a number of possibly helpful sentences $\{\boldsymbol{k}_i\}_{i=1}^{M}$ from Z, where $M \ll |Z|$, according to a relevance function f.

$$f([\boldsymbol{x};\boldsymbol{e}],\boldsymbol{k}) = E_{src}([\boldsymbol{x};\boldsymbol{e}])^T E_{tgt}(\boldsymbol{k}) \tag{1}$$

where $[;]$ indicates concatenation. E_{src} and E_{tgt} are the source and target sentence encoders that map $[\boldsymbol{x};\boldsymbol{e}]$ and $\boldsymbol{k}$ to d-dimensional vectors respectively.

To explore the performance of multiple retrieval methods, we evaluate the retrieval results on the testing set of XunZi. In other words, the retrieval sentence is compared with the expert-annotated description. The results are shown in Table 2. It can be seen that using $[\boldsymbol{x};\boldsymbol{e}]$ as a query can achieve the overall best performance, which significantly outperforms only using ancient entity $\boldsymbol{e}$. In addition, compared with traditional methods, mContriever ($[\boldsymbol{x};\boldsymbol{e}]$) also presents superiority. The main reason is that mContriever uses dense representations, which can effectively solve the out-of-vocabulary (OOV) problem. Based on these results, we choose mContriever ($[\boldsymbol{x};\boldsymbol{e}]$) for the following stages.

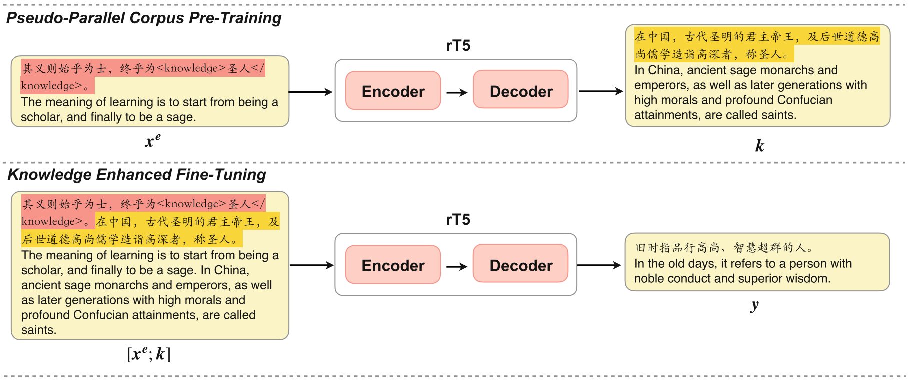

Pseudo-Parallel Corpus Pre-Training
其义则始乎为士，终乎为<knowledge>圣人</knowledge>。
The meaning of learning is to start from being a scholar, and finally to be a sage.
x^e
rT5
Encoder
Decoder
在中国，古代圣明的君主帝王，及后世道德高尚儒学造诣高深者，称圣人。
In China, ancient sage monarchs and emperors, as well as later generations with high morals and profound Confucian attainments, are called saints.
k
Knowledge Enhanced Fine-Tuning
其义则始乎为士，终乎为<knowledge>圣人</knowledge>。在中国，古代圣明的君主帝王，及后世道德高尚儒学造诣高深者，称圣人。
The meaning of learning is to start from being a scholar, and finally to be a sage. In China, ancient sage monarchs and emperors, as well as later generations with high morals and profound Confucian attainments, are called saints.
$[x^e;k]$
rT5
Encoder
Decoder
旧时指品行高尚、智慧超群的人。
In the old days, it refers to a person with noble conduct and superior wisdom.
y

Fig. 3. An illustration of pre-training and fine-tuning processes.

3.3 Pseudo-Parallel Corpus Pre-training

In light of the retrieved knowledge $\boldsymbol{k}$, we then pre-train our model rT5, which has the same encoder-decoder structure as the previous T5 [6] and its related model mT5 [7]. Initially, since ACEDG aims to generate descriptions for entities in the ancient Chinese sentence, the target entity should be highlighted. Therefore, we pre-process the input sequence by inserting special tokens that can emphasize the entity type. As the example shown in Fig. 3, given the sentence “ 为士 ,终乎为圣人 ” (*“The meaning of learning is to start from being a scholar, and finally to be a sage.”*), we highlight the entity “ 圣人 ” (*“sage”*) with its type and become “$<$ *knowledge* $>$ 圣人 $<$ */knowledge* $>$”. In this way, the original input sentence $\boldsymbol{x}$ is formulated into $\boldsymbol{x}^{\boldsymbol{e}}$.

Then we pre-train rT5 with pseudo-parallel pairs in a sequence-to-sequence learning manner. Assuming the parameter is θ, the overall pre-training objective is to model the conditional probability $p_\theta(\boldsymbol{k}|\boldsymbol{x}^e)$. Concretely, at the t-th time step, the decoder output $\boldsymbol{k}_t$ is computed with the entity-highlighted input $\boldsymbol{x}$ and the previous outputs $\boldsymbol{k}_{<t}$.

$$p_\theta(\boldsymbol{k}_t|\boldsymbol{x}^e, \boldsymbol{k}_{<t}) = \text{softmax}(W^\text{T}\boldsymbol{k}_{<t}) \tag{2}$$

where W maps $\boldsymbol{k}_{<t}$ into a vector, which can represent the probability distribution over the whole vocabulary set.

Then rT5 is pre-trained with minimizing the cross-entropy loss,

$$\mathcal{L}(\boldsymbol{x}^e, \boldsymbol{k}) = -\sum_{t=1}^{n} \text{log}p_\theta(\boldsymbol{k}_t|\boldsymbol{x}^e, \boldsymbol{k}_{<t}) \tag{3}$$

where n is the length of the external knowledge $\boldsymbol{k}$.

3.4 Knowledge Enhanced Fine-Tuning

Finally, we fine-tune the model with the expert-annotated XunZi dataset. To leverage external knowledge in this stage, modern Chinese knowledge $\boldsymbol{k}$ is simply concatenated with formatted ancient Chinese sentence $\boldsymbol{x}^e$ as the input. rT5 is further fine-tuned by minimizing the cross-entropy loss.

$$\mathcal{L}([\boldsymbol{x}^e; \boldsymbol{k}], \boldsymbol{y}) = -\sum_{t=1}^{m} \text{log}p_\theta(\boldsymbol{y}_t|[\boldsymbol{x}^e; \boldsymbol{k}], \boldsymbol{y}_{<t}) \tag{4}$$

where m is the length of the ground-truth target sequence $\boldsymbol{y}$.

4 Experiments

4.1 Experimental Setup

Datasets. We conducted experiments on the expert-annotated datasets proposed in Sect. 2. We divide XunZi dataset into training, valid and testing sets at the ratio of 6:1:3. The statistics of datasets are shown in Table 3.

Table 3. Dataset statistics. Since the scale of MengZi is too small, all data is regarded as the testing set to conduct an out-of-domain evaluation.

Dataset	All	Train	Valid	Test
XunZi	5311	3187	530	1594
MengZi	473	-	-	473

Evaluation Metrics. Both automatic evaluation metrics and human evaluation are used to access the performance of models. The automatic evaluation metrics include BLEU [8], Rouge [9] and Meteor [10]. Human Evaluation includes fluency, semantic consistency, and meaningfulness of the texts.

Implementation Details. To complete our experiment, we propose a retrieval-augmented pre-trained model called rT5. We first construct pseudo-parallel corpora based on full-text data from the book of XunZi. Divide the entire text into sentences, resulting in 8249 sentences. They are adopted to construct the pseudo-parallel corpus.

The model used in the experiment adopts a unified parameter configuration for fairness, the learning rate of the Adam optimizer is $3e^{-4}$, and the number of iterations of each cycle is 100. The maximum sequence length of the input (the entity length will not be very long, but the length of the entity description information is limited) is set to 128.

Compared Methods. To make an extensive evaluation, we choose the following strong baseline methods: 1) generation-based model: **BART** [11] and **GPT** [12]; 2) auto-encoder model, such as **UNILM** [13], which treats generation task as a sentence completion problem. Following UNILM, we also extend **RoBerta** [14] to generate sequences; 3) to extract entities, we pre-train a BERT for NER, called **XunZiBERT**, which can also be leveraged for ACEDG.

4.2 In-Domain Evaluation

Since rT5 is pre-trained on the XunZi book, we first evaluate its performance on the expert-annotated XunZi. The results are shown in Table 4.

Firstly, it can be observed that compared with strong pre-trained language models, rT5 achieves consistent improvements. Among the five models in the first part of Table 4, BART performs the best. Nevertheless, rT5 significantly outperforms BART. Secondly, compared with the naive retrieval results, i.e. mContriever ($\boldsymbol{e}$) and mContriever ($[\boldsymbol{x}; \boldsymbol{e}]$), rT5 also gains significantly. This further illustrates the superiority of our model. Finally, we can see that removing pre-training causes a consistent performance decrease, even reducing to the half scores of rT5. It is worth noting that rT5 outperforms rT5 w/o pre-train by +8.27% on BLEU-1 score, +12.84% on Rouge-1 score, and +3.73% on Meteor score. This also validates that pre-training with pseudo-parallel corpus is crucial.

4.3 Out-of-Domain Evaluation

To further validate the effectiveness of our proposed rT5 model, we test it using the entire MengZi dataset and obtain the results shown in Table 5. It is worth noting that rT5 is pre-trained on XunZi books. All baseline models and rT5 are fine-tuned on expert-annotated XunZi. After that, we directly evaluate rT5 and all baseline methods on the expert-annotated MengZi. The reason for selecting MengZi lies in its distinct philosophical perspective and unique set of entities and explanations, which effectively introduces an out-of-domain challenge.

Table 4. In-domain evaluation results compared with baseline methods. The best scores of each column are marked in bold.

Methods	XunZi					
	BLEU-1	BLEU-2	Rouge-1	Rouge-2	Rouge-L	Meteor
XunZiBERT	3.21	0.16	9.91	0.55	9.35	2.34
GPT	4.73	0.37	10.49	0.61	9.83	3.39
RoBerta	4.97	0.32	11.82	0.74	10.20	5.38
UNILM	5.37	0.35	13.16	0.71	11.36	6.49
BART	7.84	0.88	14.70	0.83	13.79	7.67
mContriever ($\boldsymbol{e}$)	4.79	0.45	6.52	0.06	5.32	2.43
mContriever ($[\boldsymbol{x}; \boldsymbol{e}]$)	7.29	1.63	13.26	2.18	10.27	9.33
rT5	**17.32**	**3.78**	**31.78**	**2.59**	**30.68**	**13.24**
w/o pre-train	9.05	1.02	18.94	0.97	18.74	9.51

Table 5. Out-of-domain evaluation results compared with baseline methods. The best scores of each column are marked in bold.

Methods	XunZi→MengZi					
	BLEU	BLEU-2	Rouge-1	Rouge-2	Rouge-l	Meteor
XunZiBERT	4.1	0.13	7.62	0.39	7.26	1.97
GPT	4.7	0.31	7.92	0.42	7.31	2.36
RoBerta	5.8	0.27	8.47	0.51	8.15	3.86
UNILM	6.3	0.28	8.94	0.47	8.75	4.35
BART	6.9	0.74	9.75	0.52	9.68	5.39
rT5	**16.52**	**3.13**	**29.51**	**2.16**	**28.18**	**9.56**
w/o pre-train	8.8	0.95	11.63	0.86	11.63	6.16

From Table 5, we can observe that rT5 achieves the best performance compared with strong baselines. Among the five baseline models, BART performs best. Then compared with it, rT5 still obtains significant improvements. By removing pre-training, the performance also consistently declines. These evaluation results demonstrate that our model achieves good performance in the out-of-domain setting, which signifies its ability to adapt to different domains within the context of ancient Chinese literature. This finding not only validates the robustness and versatility of the rT5 model, but also highlights its potential for understanding and interpreting various ancient Chinese texts.

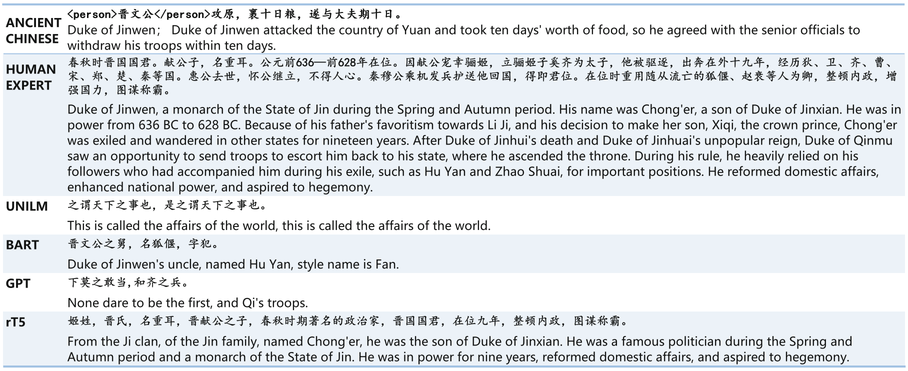
4.4 Human Evaluation

In assessing the ACEDG task, human evaluation is generally considered more dependable and trustworthy than automated evaluation measures, due to the task's more nuanced, human-like nature. The assessment criteria consist of three

Table 6. Human evaluation results, including average scores and the standard deviation of three evaluators. The best scores of each row are marked in bold.

Model	GPT	BART	UNILM	rT5
Fluency	1.6 ± 0.5	2.5 ± 0.3	1.4 ± 0.4	**3.6** ± **0.3**
Consistency	1.2 ± 0.2	1.6 ± 0.4	1.2 ± 0.2	**2.3** ± **0.4**
Meaningfulness	1.4 ± 0.2	1.5 ± 0.4	1.1 ± 0.1	**3.4** ± **0.4**

distinct dimensions, each scored on a scale of 1 to 5. The fluency dimension appraises the grammatical smoothness of the produced text. The consistency dimension measures the degree to which the translated sentence preserves the original sentence's content. The meaningfulness dimension assesses whether the words in the output sentence convey substantive meanings. The ultimate human evaluation score is computed by taking the average of the scores assigned by all human evaluators. The results are shown in Table 6. It can be observed that among all three dimensions, rT5 generates entity descriptions with the best quality. The human evaluation further verifies the effectiveness of our approach.



ANCIENT CHINESE	<person>晋文公</person>攻原，裹十日粮，遂与大夫期十日。 Duke of Jinwen； Duke of Jinwen attacked the country of Yuan and took ten days' worth of food, so he agreed with the senior officials to withdraw his troops within ten days.
HUMAN EXPERT	春秋时晋国国君。献公子，名重耳。公元前636—前628年在位。因献公宠幸骊姬，立骊姬子奚齐为太子，他被驱逐，出奔在外十九年，经历狄、卫、齐、曹、宋、郑、楚、秦等国。惠公去世，怀公继立，不得人心。秦穆公乘机发兵护送他回国，得即君位。在位时重用随从流亡的狐偃、赵衰等人为卿，整顿内政，增强国力，图谋称霸。 Duke of Jinwen, a monarch of the State of Jin during the Spring and Autumn period. His name was Chong'er, a son of Duke of Jinxian. He was in power from 636 BC to 628 BC. Because of his father's favoritism towards Li Ji, and his decision to make her son, Xiqi, the crown prince, Chong'er was exiled and wandered in other states for nineteen years. After Duke of Jinhui's death and Duke of Jinhuai's unpopular reign, Duke of Qinmu saw an opportunity to send troops to escort him back to his state, where he ascended the throne. During his rule, he heavily relied on his followers who had accompanied him during his exile, such as Hu Yan and Zhao Shuai, for important positions. He reformed domestic affairs, enhanced national power, and aspired to hegemony.
UNILM	之谓天下之事也，是之谓天下之事也。 This is called the affairs of the world, this is called the affairs of the world.
BART	晋文公之舅，名狐偃，字犯。 Duke of Jinwen's uncle, named Hu Yan, style name is Fan.
GPT	下莫之敢当,和齐之兵。 None dare to be the first, and Qi's troops.
rT5	姬姓，晋氏，名重耳，晋献公之子，春秋时期著名的政治家，晋国国君，在位九年，整顿内政，图谋称霸。 From the Ji clan, of the Jin family, named Chong'er, he was the son of Duke of Jinxian. He was a famous politician during the Spring and Autumn period and a monarch of the State of Jin. He was in power for nine years, reformed domestic affairs, and aspired to hegemony.

Fig. 4. An illustration of the case study.

4.5 Case Study

To further evaluate the proposed model, a case study is conducted. The details are depicted in Fig. 4. These models are tested by generating an explanation for the historical entity "晋文公" *(Duke of Jinwen)*. For benchmarking, we use a human expert's explanation as a reference. Upon comparison, it can be observed that the rT5 model's output is the most accurate and comprehensive, closely matching the reference description provided by the human expert. The UNILM, BART, and GPT models' outputs are not entirely correct, lacking depth and completeness compared with rT5. The superior performance of our rT5 model over existing sequence-to-sequence models demonstrates its potential in facilitating the understanding and analysis of ancient Chinese texts.

5 Related Works

NLP Applications for Ancient Chinese. Previous studies have gained great success by applying NLP techniques to ancient Chinese. To name a few, Chang *et al.* leverage local features for named entity recognition [15]. Li *et al.* [2] propose an approach that combines conditional variational autoencoder (CVAE) and adversarial training for Chinese poem generation. To better produce smooth poetry that fits the topic, Yang *et al.* [16] explore unsupervised machine translation (UMT) to generate classical Chinese poems from the vernacular, which allows the controlling over the semantics of generated poems. Other interesting works include couplet generation [3,17], a part of traditional Chinese culture and formatted as two sentences with symmetrical meanings. In this work, we focus on the entity description of ancient entities, which is meaningful for historical and cultural diffusion.

Retrieval-Based Generation. To utilize external knowledge, many retrieval-based generation works have been proposed. Guu *et al.* [18] propose to augment the pre-training of language model based on retrieval. Wang *et al.* [19] propose a new framework using retrieval methods to enhance the pre-training and fine-tuning of general knowledge generation. The prototype candidate sentence is retrieved by concept matching and used as an auxiliary input. In this paper, we use a general-purpose dense retriever based on the dual-encoder architecture of mContriever [5] to retrieve Wikipedia content according to the current context content, entity type, entity information. This aims to generate a large number of suitable pseudo-parallel corpus to enhance pre-training.

6 Conclusion

In this paper, we define a new task (i.e. ACEDG) and propose two expert-annotated datasets (i.e. XunZi and MengZi), for promoting Chinese culture and history. To tackle this task, a retrieval-augmented pre-trained model, i.e. rT5, is proposed. Specifically, we first adopt the retrieval technique for building pseudo pair of ancient Chinese entity and modern Chinese description. Then, the pseudo-parallel corpus is leveraged to pre-train our model, which incorporates external knowledge. Finally, rT5 is fine-tuned on the expert-annotated dataset with the help of external knowledge. Experimental results under various metrics show that our method can generate higher-quality entity descriptions. The rT5 model's success in this study highlights the potential of leveraging information retrieval-enhanced techniques for dataset construction and domain-specific performance. Future research can explore the application of these techniques in other domains and languages to further enhance the understanding of ancient Chinese.

References

1. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**(1), 50–70 (2020)
2. Li, J.: Generating classical Chinese poems via conditional variational autoencoder and adversarial training. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3890–3900. Association for Computational Linguistics, Brussels, Belgium, October–November 2018
3. Wang, Y., Zhang, J., Zhang, B., Jin, Q.: Research and implementation of Chinese couplet generation system with attention based transformer mechanism. IEEE Trans. Comput. Soc. Syst. (2021)
4. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Izacard, G., et al.: Unsupervised dense information retrieval with contrastive learning (2021)
6. Raffel, C.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1), 5485–5551 (2020)
7. Xue, L.: mT5: a massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934 (2020)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, July 2002
9. Lin, C.-Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona, Spain, July 2004
10. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan, June 2005
11. Lewis, M.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
12. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
13. Dong, L.: Unified language model pre-training for natural language understanding and generation. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
14. Liu, Y.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
15. Chang, Y., Kong, L., Jia, K., Meng, Q.: Chinese named entity recognition method based on BERT. In: 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), pp. 294–299 (2021)
16. Yang, Z., et al.: Generating classical Chinese poems from vernacular Chinese. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. In: Conference on Empirical Methods in Natural Language Processing, vol. 2019, p. 6155. NIH Public Access (2019)

17. Yuan, S., Zhong, L., Li, L., Zhang, R.: Automatic generation of Chinese couplets with attention based encoder-decoder model. In: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 65–70. IEEE (2019)
18. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International Conference on Machine Learning, pp. 3929–3938. PMLR (2020)
19. Wang, H.: Retrieval enhanced model for commonsense generation. arXiv preprint arXiv:2105.11174 (2021)

Probing Bilingual Guidance for Cross-Lingual Summarization

Dawei Zhu[1,2], Wenhao Wu[1,2], and Sujian Li[1,2](✉)

[1] School of Computer Science, Peking University, Beijing, China
{dwzhu,waynewu,lisujian}@pku.edu.cn

[2] National Key Laboratory for Multimedia Information Processing, Peking University, Beijing, China

Abstract. Bilingual guidance as an augmentation method is often applied in Seq2Seq cross-lingual summarization (CLS) to improve its performance. In this paper, we empirically explore bilingual guidance from the encoder/decoder side, as well as the perspective of representation alignment. We examine the respective benefits of their influence on CLS, and how this benefit varies with regard to guidance quality. Toward this, we devise a probing model to incorporate bilingual guidance, and construct a probing dataset containing bilingual document-summary pairs. We also comprehensively examine the representation alignment strategy for utilizing bilingual guidance in CLS. Empirical results robustly substantiate the impact of bilingual guidance on both the encoder and decoder aspects, and affirm the efficacy of representation alignment in the context of CLS. We also have some inspiring discoveries that encoder side guidance is more sensitive to guidance quality, and that representation alignment greatly benefits low-resource settings. Our code is publicly available at https://github.com/dwzhu-pku/BiGuid.

Keywords: cross-lingual summarization · bilingual guidance · representation alignment

1 Introduction

Cross-lingual Summarization (CLS) is a task that transforms a document in one language into a summary in another language [12]. This novel task enables a quick grasp of information manifested in foreign languages, which is gaining importance under the background of globalization. Initially, the task is tackled in a pipeline fashion, namely first-translate-then-summarize or first-summarize-then-translate [12,17,26,27]. Later, the use of sequence-to-sequence (Seq2Seq) training [24] brought significant improvements over traditional pipeline methods by simply training with one language as input and another language as output [7]. In this case, the translated documents or summaries often served as augmented data for a Seq2Seq model. This kind of bilingual guidance has proven useful in CLS [1,2,7,22,33], and remains advantageous even in the era of large

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 749–760, 2023.
https://doi.org/10.1007/978-3-031-44693-1_58

language models. However, there is still no comprehensive exploration of how bilingual guidance benefits cross-lingual summarization.

First, we examine the application of bilingual guidance across different modules, considering varying levels of quality. To achieve this, bilingual parallel corpus is necessary. With the availability of bilingual news websites and advancements in machine translation, human-calibrated and machine-translated parallel resources for CLS corpora (i.e., translated input documents in *target* language and translated target summaries in *source* language) have become feasible. Then, we address the following research questions: *RQ.1*: What are the individual benefits of bilingual guidance from the encoder and decoder perspectives, and can their combination enhance model performance? *RQ.2*: How does the quality of guidance, specifically the parallel resources, influence model performance?

Additionally, our investigation focuses on the influence of bilingual representation alignment on cross-lingual summarization performance. Representation alignment is a crucial strategy in cross-lingual transfer learning, aiming to obtain consistent representations for sentences expressing the same semantic meaning across different languages, i.e. sentences in different languages expressing the same semantic meaning should have similar representations [4,6,20,29]. While its effectiveness has been established in tasks such as cross-lingual Natural Language Inference (NLI) [32] and Semantic Textual Similarity (STS) [29], its potential impact on natural language generation tasks, particularly in CLS, remains underexplored. In this paper, equipped with the parallel resources through human and machine translation, a comprehensive inspection of representation alignment for CLS is made feasible. Specifically, we explore two research questions in this line: *RQ.3*: How will representation alignment benefit cross-lingual summarization? *RQ.4*: How will alignment effect vary with regard to different training data size?

Concentrating on the aforementioned four research questions, this paper presents an empirical study concerning the influence of bilingual guidance and representation alignment for CLS under the Seq2Seq framework. For the convenience of this analysis, we construct a probing dataset called BiNews, and devise a probing model named BiGuid. BiNews is a high-quality summarization dataset containing bilingual document-summary pairs collected from Bilingual news websites. BiGuid is a transformer-based summarization model capable of incorporating bilingual guidance in both encoder and decoder sides. The empirical experiments will show how bilingual guidance and representation alignment influence CLS under the Seq2Seq framework.

In summary, the contributions of our work are as follows: 1) We consolidate the effectiveness of bilingual guidance on both the encoder and decoder sides, and affirm the efficacy of representation alignment in the context of cross-lingual summarization through comprehensive experiments. 2) To reveal the importance of guidance quality for cross-lingual summarization, we construct a bilingual news dataset containing human-calibrated and machine-translated parallel resources. 3) We also have some inspiring discoveries that encoder side guidance is more sensitive to guidance quality, and that representation alignment greatly benefits low-resource settings.

2 Model Design

In this section we introduce the probing model **BiGuid**, which is designed to incorporate bilingual guidance in both encoder and decoder sides for empirical studies. Its architecture is illustrated in Fig. 1, accompanied by two of its variants (b) **EncGuid** and (c) **DecGuid**, which respectively stands for the structure of BiGuid when there are only encoder-side or decoder-side guidance available.

For encoder-side guidance, we employ two encoders with tied parameters to process the input documents in the source and target languages (x and x^G), respectively. The cross-attention block in decoder attends to both encoder's output. For decoder-side guidance, we adopt the multi-task learning strategy and train the decoder to summarize in target or source language (y and y^G) given the corresponding start token (s and s^G).

2.1 Design for Encoder-Side Guidance

For encoder-side guidance, two encoders are adopted to separately encode x and x^G into their contextual representation h and h^G. Each encoder is composed of a stack of transformer layers [25]. Following previous work [2], we apply a unified BPE [21] dictionary and share two encoders' parameters to enhance the isomorphism of both contextual representations.

The decoder also comprises of a stack of transformer layers, with each layer including a self attention block, a cross attention block, and a feed forward block. It attends to both encoders' outputs to produce target summary y. Concretely, assume that the self attention block produces contextual representation o, the cross attention first attends o to the encoded bilingual representations h and h^G, respectively:

$$v, v^G = \text{MultiHead}(h, h, o), \text{MultiHead}(h^G, h^G, o)$$

where $\text{MultiHead}(x, y, z)$ represents applying x, y and z as key, value and query for multi-head attention, respectively. The result v and v^G is fused through a linear layer before passed to the feed forward block.

2.2 Design for Decoder-Side Guidance

We apply multi-task learning to take advantage of decoder-side guidance. We first teach the model to generate the reference summary y with Negative Log-Likelihood (NLL) loss:

$$\mathcal{L}_T = -\sum_t \log P(y_t | y_{<t}, x, x^G, s) \tag{1}$$

where t is the generation step of decoding, s is the start token for generating text in target language. To optimize reference summary generation, we also generate the translated summary y^G as an auxiliary task, and its NLL loss $\mathcal{L}^G$ is computed analogously. The overall training loss is defined as: $\mathcal{L} = \mathcal{L}_T + \lambda \mathcal{L}^G$, where $\lambda \in (0, 1]$ controls the weights.

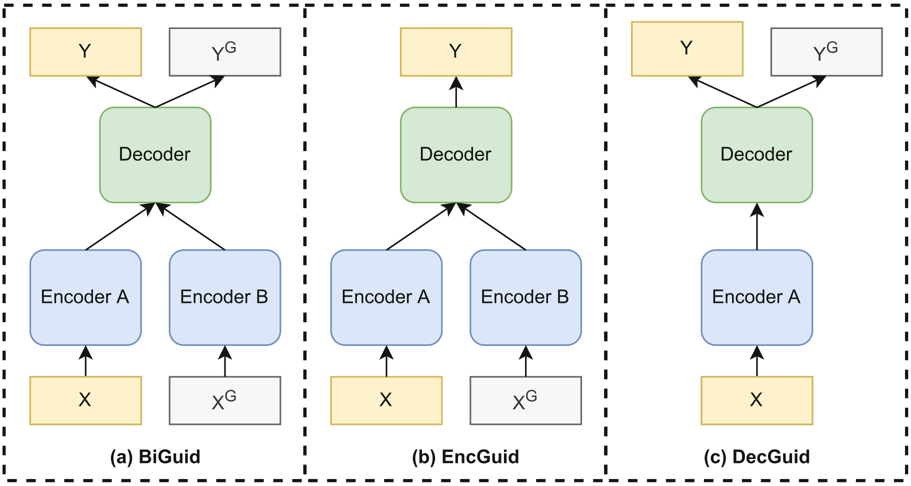


Fig. 1. The Architecture of BiGuid

2.3 Loss Function for Representation Alignment

In addition to the NLL loss defined in Eq. 1, an alignment loss term $\mathcal{L}_{align}$ is applied to explore the effect of sentence-level representation alignment in CLS. Specifically, given the contextual representations h and h^G of semantically parallel documents x and x^G, three types of loss functions are adopted to explore representation alignment:

Squared Error minimizes Euclidean distance of h and h^G:

$$\mathcal{L}_{SE} = ||h - h^G||_2 \tag{2}$$

KL Divergence measures distance between probability distributions. For discrete distributions P and Q, it is as follows:

$$\mathcal{D}_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

To utilize KL divergence alignment, h and h^G is passed through softmax function to obtain the normalized version $\hat{h} = \text{softmax}(h)$ and $\hat{h^G} = \text{softmax}(h^G)$, upon which the KL divergence loss is calculated as:

$$\mathcal{L}_{KL} = \mathcal{D}_{KL}(\hat{h}||\hat{h^G}) \tag{3}$$

Npair Loss [23] is a contrastive learning strategy to minimize the representation distance between positive pairs and maximize that of negative pairs. Concretely, in a batch of 1 positive sample and N-1 negative samples $\{h, h^+, h_1^-, h_2^-, ..., h_{N-1}^-\}$ for anchor point h, the npair loss is:

$$\mathcal{L}_{Npair} = -\log \frac{exp(h^T h^+)}{exp(h^T h^+) + \sum_{i=1}^{N-1} exp(h^T h_i^-)} \tag{4}$$

In practice, we treat h^G as positive for h, and embeddings for other input text in the same batch as negative.

3 Probing Dataset Construction

For an empirical study concerning bilingual guidance, a CLS corpus with translated parallel resource is indispensable. To this end, we construct a dataset named BiNews, which contains 247,157 bilingual news and headlines written in both Chinese and English in parallel. These news and headlines are collected from several bilingual news websites and are mostly calibrated by human.[1]

Data cleaning and other preprocessing procedures are performed, before we randomly split BiNews into 241,157/3,000/3,000 for training, validation and testing, respectively. On average, each English document/summary contains 593.83/16.42 words and each Chinese document/summary contains 1,207.80/37.27 characters. Detailed statistics about the BiNews dataset can be found in Table 1.

Figure 2 is an example of BiNews. The first four blocks are a sample from BiNews, including the English and Chinese documents and summaries, which are semantically parallel. The fifth block shows a Chinese summary translated from the parallel English Summary (block 2) using google-translation. It can be seen that our collected Chinese summary (block 4) outperforms machine translation in quality, since the latter omits the sense of *calming effect*. BiNews makes it possible to compare machine-translated or human-calibrated bilingual guidance to study the impact of guidance quality.

Table 1. Statistics of the BiNews dataset. The table presents the average number of words per document, sentences per document, and words per sentence for English and Chinese documents, as well as English and Chinese summaries in the train, valid, and test sets.

Model	#word/document			#sentence/document			#word/sentence		
	train	valid	test	train	valid	test	train	valid	test
English Document	593.83	596.54	597.75	17.67	17.07	17.57	33.61	34.94	34.01
English Summary	16.42	16.38	16.43	1.05	1.05	1.05	15.62	15.63	15.58
Chinese Document	1207.8	1213.15	1214.6	29.15	29.26	29.28	41.43	41.46	41.48
Chinese Summary	37.27	37.27	37.1	1.46	1.44	1.45	25.59	25.88	25.51

[1] News websites such as China Daily, VOA, New York Time, contain bilingual news to benefit language learners. The translation part of bilingual news is typically curated by human reviewers.

①	**English Document:** However, a new study conducted by the University of Limerick in Ireland found that resistance exercise training, like weightlifting, may actually help soothe anxiety.
②	**English Summary:** Research finds that weightlifting has surprisingly calming effects.
③	**Chinese Document:** 然而，爱尔兰利默里克大学进行的一项新研究发现，像举重一样的阻力运动训练实际上可能有助于缓解焦虑。(However, a new study carried out by University of Limerick in Ireland found that, resistance exercise like weight lifting maybe actually help relieve anxiety.)
④	**Chinese Summary:** 研究发现举重具有惊人的镇静效果。 (Reaches find that weightlifting has surprisingly calming effect.)
⑤	**Chinese Summary (Machine Translated):** 研究发现，举重效果令人惊讶。(Reaches find that the effect of weightlifting is surprising.)

Fig. 2. Example of BiNews dataset. The first four blocks show a sample of BiNews. The fifth block shows a Chinese summary translated with machine.

Table 2. Experiment results on BiNews for Chinese-to-English (Zh2En) and English-to-Chinese (En2Zh) summarization. *EncGuid* and *DecGuid* respectively stands for guidance in encoder and decoder side. *Machine* and *oracle* indicate guidance of machine-translated or human-calibrated quality. Underline means the highest score among EncGuid and DecGuid. Highest score of a column is denoted in **bold**.

Model	Zh2En			En2Zh		
	R-1	R-2	R-L	R-1	R-2	R-L
Base	31.12	11.38	25.85	39.44	23.94	32.82
EncGuid (machine)	32.99	12.71	28.07	40.77	25.70	33.79
EncGuid (oracle)	41.64	21.16	36.39	43.99	29.40	37.30
DecGuid (machine)	35.20	14.29	29.79	40.33	25.05	33.06
DecGuid (oracle)	36.05	15.01	30.82	42.97	27.75	35.72
BiGuid	**42.74**	**21.91**	**37.15**	**45.28**	**30.56**	**38.50**

4 Experiments

In this section, we conduct comprehensive experiments on BiNews to study the influence of bilingual guidance and representation alignment. We also apply BiGuid to existing CLS datasets En2ZhSum and Zh2EnSum to compare its performance with previous works. Following the convention, the quality of generated summaries are evaluated using ROUGE-1, 2 and L [14]. We use a unified BPE vocabulary with a size of 15,000. For our model, the number of both encoder and decoder layers is 6, the hidden size is 512, and the number of heads in multi-head attention is 8. Fairseq [18] framework is used to implement all the models above.

We apply Adam optimizer [10] with $\beta_1 = 0.9$, $\beta_2 = 0.998$, and $\epsilon = 10^{-9}$. We set dropout rate to 0.1 and warmup steps to 8000.

4.1 Bilingual Guidance Analysis

Table 3. Performance of BiGuid using different alignment loss term with different training set size.

Data	Loss	Zh2En			En2Zh		
		R-1	R-2	R-L	R-1	R-2	R-L
10%	NLL	20.53	4.73	17.01	29.76	14.92	23.50
	NLL+SE	**26.69**	**7.37**	**21.75**	29.72	14.86	23.54
	NLL+KL	26.38	7.10	21.64	29.58	14.82	23.43
	NLL+Npair	24.92	6.73	20.71	**30.44**	**15.37**	**23.97**
30%	NLL	37.03	16.87	32.14	38.14	23.71	31.62
	NLL+SE	**37.54**	16.96	**32.61**	**38.77**	**24.13**	**32.12**
	NLL+KL(△)	36.81	16.55	31.76	38.40	24.07	31.79
	NLL+Npair	37.51	**17.43**	32.57	38.34	23.98	31.87
100%	NLL	41.59	21.02	36.22	44.24	29.78	37.52
	NLL+SE	41.60	21.31	36.48	44.34	29.74	37.43
	NLL+KL(△)	41.52	21.07	36.10	44.11	**29.93**	37.81
	NLL+Npair	**41.89**	**21.21**	**36.56**	**44.34**	29.87	**37.81**

Effectiveness of Bilingual Guidance with Different Guidance Quality. We first examine the CLS performance with bilingual guidance of different quality on encoder and decoder side, independently. The experiment results are exhibited in Table 2. *Base* denotes the BiGuid model without guidance, where it degrades into a vanilla transformer architecture. *EncGuid/DecGuid* are BiGuid with only encoder/decoder side guidance, as mentioned before. *Machine* and *oracle* stands for guidance of different quality. The former is obtained through machine translation techniques, specifically the google-translation api. The latter is human-calibrated translation unique to our bilingual news dataset. Firstly, it is unsurprising that models under oracle guidance yields better results than their machine-guided counterparts, obtaining an average performance gain of more than 2 ROUGE points in most cases (*RQ.2*). Secondly, both EncGuid and DecGuid surpass the guidance-less Base model by a large margin, even with only machine translation for guidance. Hence it is safe to conclude that bilingual guidance in both encoder and decoder side is beneficial for CLS (*RQ.1*). Thirdly, it is note-worthy that encoder side guidance is more sensitive to guidance quality, as performance gain of oracle guidance is much larger than that of decoder. We speculate that under the oracle setting, the model can simply perform monolingual summarization, which requires not the ability of cross-lingual transfer and is hence easier to manage.

Compatibility of Encoder and Decoder Side Guidance. With the independent effectiveness of bilingual guidance in encoder/decoder side being verified, we further explore the compatibility of EncGuid and DecGuid, i.e. whether model performance can be further improved by utilizing guidance signal from both sides (*RQ.1*). The last block of Table 2 presents ROUGE scores of BiGuid model combining encoder and decoder side guidance. In Chinese-to-English summarization, BiGuid achieves an improvement of 1.10/0.75/0.76 points on R-1/R-2/R-L compared with the best-performing EncGuid model before. In the reverse direction, the performance gain is 1.19/1.16/1.20 for R-1/R-2/R-L, which is a significant improvement as well. Hence, combining guidance from both sides can yields further improvement.

4.2 Representation Alignment Analysis

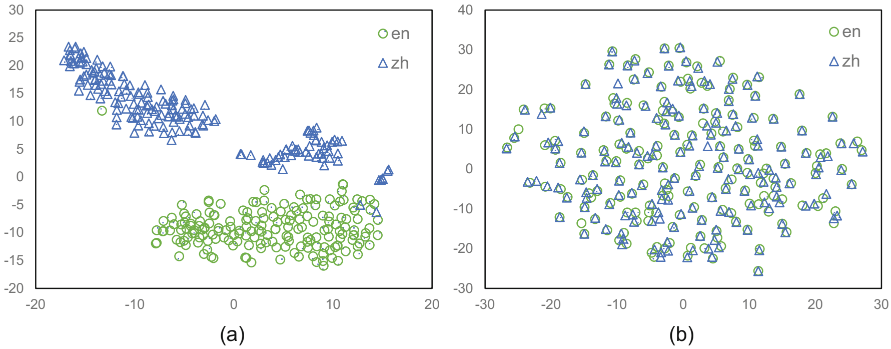


Fig. 3. Sentence representation extracted from model trained without (a) and with (b) the alignment loss term.

Effectiveness of Representation Alignment for CLS. Table 3 presents model performance using different alignment loss term with 10%/30%/100% training data. In most cases, alignment loss term elevates model performance (*RQ.3*), with the two exceptions marked out with △. Additionally, it is observed that different alignment strategies achieve enhancement to different extent under different data settings. *Npair* alignment strategy delivers the best performance gain under full data setting, while *Squared Error* is also competitive when data size decreases. Furthermore, we visualize the representations of parallel English and Chinese sentences using T-SNE, as in Fig. 3. It can be seen that alignment strategy improves isomorphism of sentence representations in different languages.

Alignment Benefits Low-Resource Settings. By comparing performance gain of certain alignment strategy given different training data size, trend is clear that as the data size increases, performance gain from representation alignment will decline (*RQ.4*). Take Npair loss as an example, with 10% of training data, this alignment strategy averagely receives 3.36 ROUGE points boost for Chinese-to-English (Zh2En) CLS, and 0.53 ROUGE points boost for English-to-Chinese (En2Zh) CLS, whereas with 100% of training data, the average performance gains drop to 0.27 and 0.16 for Zh2En and En2Zh, respectively. We speculate that representation alignment benefits low-resource settings by improving the isomorphism of embedding space for different languages, which can also be compensated through large amount of training data.

4.3 Comparison with Other Non-pretrained Models

Table 4. Comparison with baselines and previous works on En2ZhSum and Zh2EnSum. With a simple architecture, BiGuid achieves comparable results with previous works.

Model	En2ZhSum			Zh2EnSum		
	R-1	R-2	R-L	R-1	R-2	R-L
TNCLS	36.82	18.72	33.20	38.85	21.93	35.05
CLS+MT	40.23	22.32	36.59	40.25	22.57	36.21
CLS+MS	38.25	20.20	34.76	40.34	22.65	36.39
ATS-A	40.47	22.21	36.89	40.68	24.12	36.97
MAP-REC	38.12	16.76	33.86	40.97	23.20	36.96
VHM	40.98	**23.07**	37.12	**41.36**	**24.64**	**37.15**
BiGuid	**41.32**	22.57	**37.52**	41.02	22.04	36.29

In addition to the empirical analysis on our dataset, BiGuid is also applied to previous CLS datasets En2ZhSum and Zh2EnSum [33] to examine the potential of bilingual guidance and representation alignment in real settings. Both datasets are constructed through round-trip translation. We adopt google-translation to obtain parallel resources for guidance.

Table 4 presents the results of BiGuid on these two datasets, along with previous works. For fair comparison, we only compare BiGuid with no-pretrained models. TNCLS is the baseline model adopting a vanilla transformer architecture. CLS+MT and CLS+MS [33] are two multi-task strategies which simultaneously trains CLS with MT or MLS. ATS-A [34] is a pointer-generator network utilizing translation patterns in CLS. MAP-REC [2] designs a multi-task framework and reconstruction loss term to learn a mapping of representation between languages. VHM [13] is a hierarchical model based on conditional variational auto-encoder achieving states-of-the-art in non-pretrained settings.

We find that BiGuid, though being a simple architecture, gives large improvements over baselines, surpasses most of the competitive models, and achieves comparable results with SOTA in non-pretrained settings. These impressive results demonstrates the potential of incorporating bilingual guidance and representation alignment in cross-lingual summarization.

5 Related Work

5.1 Cross-Lingual Summarization

Cross-lingual summarization (CLS) involves generating summaries in one language from documents in another language. Earlier research in this area mainly focused on pipeline methods that involve translation and summarization or summarization and translation [12,17,26,27]. With the appearance of large-scale CLS datasets [33] and the rise of sequence-to-sequence [24] training paradigm, there has been a shift in research attention towards end-to-end CLS models, which encompass various frameworks [28], such as multitask learning [1,2,13], knowledge distillation [16], resource-enhanced [9,34], and pretraining [3,30]. It is note worth that a significant portion of these work incorporate bilingual guidance in some way. Our work provides a comprehensive analysis of how such guidance can benefit CLS.

5.2 Cross-Lingual Transfer Learning

Cross-lingual transfer learning is a technique [19,20] that seeks to transfer knowledge from a source language with abundant corpus resources to a target language with limited corpus resources. The primary objective of this approach is to address the performance gap [31] that exists across languages, whereby model performance is superior for rich-resource languages, such as English, but inferior for low-resource languages. This technique is especially beneficial for rapidly transferring NLP systems to new languages.

Representation alignment is a critical strategy employed in cross-lingual transfer learning, which involves obtaining an isomorphic representation across various languages using parallel data. Wang et al. (2021) [29] align sentence representations with Dual Momentum Contrast to improve results on cross lingual semantic textual similarity task. By introducing new training data and new training methods, a series of pretrained models has also been proposed, including Unicoder [8], XLM [11], XLM-R [5], VECO [15], VECO 2.0 [32]. However, these methods are tailored for natural lanuage understanding tasks such as NLI and NER, and the effectiveness of Representation alignment in generation tasks remains underexplored. Our study collects parallel language resources and provides a comprehensive analysis of representation alignment for CLS.

6 Conclusions

In this paper, we empirically explore four research questions regarding bilingual guidance for CLS. To this end, we design a probing model and collect a probing dataset. Comprehensive experiments verify the effect of bilingual guidance in both encoder and decoder sides, and substantiate the efficacy of representation alignment for CLS. We also have some inspiring discoveries that encoder side guidance is more sensitive to guidance quality, and that alignment greatly benefits low-resource settings. Comparison with previous works further illustrates the promising future of using bilingual guidance and representation alignment for cross-lingual summarization.

Acknowledgement. We thank the anonymous reviewers for their helpful comments on this paper. This work was partially supported by National Key R&D Program of China (No. 2022YFC3600402) and National Social Science Foundation Project of China (21&ZD287).

References

1. Bai, Y., Gao, Y., Huang, H.: Cross-lingual abstractive summarization with limited parallel resources. In: Proceedings of ACL (2021)
2. Cao, Y., Liu, H., Wan, X.: Jointly learning to align and summarize for neural cross-lingual summarization. In: Proceedings of ACL (2020)
3. Chi, Z., et al.: mT6: multilingual pretrained text-to-text transformer with translation pairs. In: Proceedings of EMNLP (2021)
4. Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: Unsupervised cross-lingual representation learning for speech recognition. arXiv preprint arXiv:2006.13979 (2020)
5. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116 [cs] (2020)
6. Conneau, A., et al.: XNLI: evaluating cross-lingual sentence representations. In: EMNLP (2018)
7. Duan, X., Yin, M., Zhang, M., Chen, B., Luo, W.: Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In: Proceedings of ACL (2019)
8. Huang, H., et al.: Unicoder: a universal language encoder by pre-training with multiple cross-lingual tasks. In: Proceedings of EMNLP (2019)
9. Jiang, S., Tu, D., Chen, X., Tang, R., Wang, W., Wang, H.: ClueGraphSum: let key clues guide the cross-lingual abstractive summarization (2022). https://doi.org/10.48550/arXiv.2203.02797
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
11. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv:1901.07291 [cs] (2019)
12. Leuski, A., Lin, C., Zhou, L., Germann, U., Och, F.J., Hovy, E.H.: Cross-lingual C*ST*RD: English access to Hindi information. ACM Trans. Asian Lang. Inf. Process. (2003)
13. Liang, Y., et al.: A variational hierarchical model for neural cross-lingual summarization. In: Proceedings of ACL (2022)

14. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out (2004)
15. Luo, F., et al.: VECO: variable and flexible cross-lingual pre-training for language understanding and generation. In: Proceedings of ACL (2021)
16. Nguyen, T.T., Luu, A.T.: Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation. In: Proceedings of AAAI (2022)
17. Orasan, C., Chiorean, O.A.: Evaluation of a cross-lingual Romanian-English multi-document summariser. In: Proceedings of LREC (2008)
18. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: Proceedings of AACL (2019)
19. Prettenhofer, P., Stein, B.: Cross-lingual adaptation using structural correspondence learning. ACM Trans. Intell. Syst. Technol. (2011)
20. Ruder, S., Vulić, I., Søgaard, A.: A survey of cross-lingual word embedding models. J. Artif. Int. Res. (2019)
21. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of ACL (2016)
22. Shen, S.Q., Chen, Y., Yang, C., Liu, Z.Y., Sun, M.S.: Zero-shot cross-lingual neural headline generation. IEEE/ACM Trans. Audio Speech Lang. Proc. (2018)
23. Sohn, K.: Improved deep metric learning with multi-class N-pair loss objective. In: Proceedings of NeurIPS (2016)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of NeurIPS (2014)
25. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)
26. Wan, X.: Using bilingual information for cross-language document summarization. In: Proceedings of ACL (2011)
27. Wan, X., Li, H., Xiao, J.: Cross-language document summarization based on machine translation quality prediction. In: Proceedings of ACL (2010)
28. Wang, J., et al.: A survey on cross-lingual summarization. arXiv:2203.12515 [cs] (2022)
29. Wang, L., Zhao, W., Liu, J.: Aligning cross-lingual sentence representations with dual momentum contrast. In: Proceedings of EMNLP (2021)
30. Xu, R., Zhu, C., Shi, Y., Zeng, M., Huang, X.: Mixed-lingual pre-training for cross-lingual summarization. In: Proceedings of AACL (2020)
31. Yang, H., Chen, H., Zhou, H., Li, L.: Enhancing cross-lingual transfer by manifold mixup (2022)
32. Zhang, Z.R., Tan, C., Huang, S., Huang, F.: VECO 2.0: cross-lingual language model pre-training with multi-granularity contrastive learning (2023)
33. Zhu, J., et al.: NCLS: neural cross-lingual summarization. In: Proceedings of EMNLP (2019)
34. Zhu, J., Zhou, Y., Zhang, J., Zong, C.: Attend, translate and summarize: an efficient method for neural cross-lingual summarization. In: Proceedings of ACL (2020)

Accurate, Diverse and Multiple Distractor Generation with Mixture of Experts

Fanyi Qu[1,2], Che Wang[1,2], and Yunfang Wu[1,2](✉)

[1] National Key Laboratory for Multimedia Information Processing,
Peking University, Beijing, China
{fanyiqu,2000013110,wuyf}@pku.edu.cn
[2] School of Computer Science, Peking University, Beijing, China

Abstract. Given the background passage, question and answer, Distractor Generation (DG) aims to generate several incorrect options to confuse readers, which is an essential composition to build multiple choice question data. Most of the existing works apply naive methods to obtain multiple outputs with the sacrifice of the generation quality. In this paper, we propose an end-to-end one-to-many generation structure with mixture of experts (MoE) for DG, and explore how different data-to-expert routing strategies of MoE influence the performance of one-to-many generation models. Concretely, the model's encoder calculates token-level attention vectors to mark important tokens from the source sequence, and the decoder generates multiple results with the guidance of the local attention. Moreover, we propose a minimal loss assignment mechanism and a stable routing strategy for diversity generation. Experimental results demonstrate that our proposed method is able to generate multiple distractors with good interpretability, which greatly outperforms the existing state-of-the-art DG models in quality, and achieves satisfactory lexical and semantic diversity.

Keywords: Distractor generation · Mixture of Experts · Routing strategy

1 Introduction

Multiple Choice Question (MCQ) data is extensively applied to reading comprehension assessment in the educational field, which consists of four necessary components: passage, question, answer and several distractors. In recent years, much work has devoted to question generation and answer extraction tasks, while Distractor Generation (DG), which generates multiple incorrect answers to confuse readers, is less addressed.

Compared to other text generation tasks, DG is more challenging due to the following reasons: 1) It is more difficult to generate an *incorrect* content than a correct one. 2) As a one-to-many generation task, the DG model is required to

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 761–773, 2023.
https://doi.org/10.1007/978-3-031-44693-1_59

generate *multiple* distractors given one source passage. 3) To effectively confuse readers, the generated sequences should be semantic-related with the passage and question (*good quality*) and are different from each other (*high diversity*).

Table 1. The BLEU-4 scores of 3 generated distractors by HSA [6], HCA [19], HMD-Net [12] and our proposed model.

	Distractor 1	Distractor 2	Distractor 3
HSA	6.43	5.17	4.59
HCA	7.01	5.51	4.88
HMD-Net	7.66	6.37	5.33
Ours	9.52	9.12	9.59

In order to generate multiple results with one source sequence, the common strategy of previous work is to perform an unequal generation process for different results with the sacrifice of generation quality. A typical method is beam-search with Jaccard Distance [6,13,19]. These works generate multiple results in descending likelihood with a large beam size, where the Jaccard Distance between different outputs is set to be greater than a threshold, and then select out top K results. This method suffers from a similar problem: with the increase of the number of generated results, the generation quality declines sharply, as shown in Table 1. Besides, they are lack of interpretability to show whether they pay attention to different contents of the passage and in what way they produce different outputs. As a result, these works just achieve lexical diversity rather than semantic diversity [17].

Some recent works adopt Mixture of Experts (MoE) [7] to address the one-to-many generation task [2]. As for DG, Some work [17] employs the MoE method with a pipeline structure, which first extracts key sentences from the input passage and then produces multiple outputs with the key sentences as input. Since there are no supervised labels for key content extraction in the dataset, empirical rules with well-designed hyper-parameters are constructed to annotate key sentences in the passage, which are highly correlated with the distribution of training data and thus hard to be generalized in real-world applications.

In this paper, we propose a novel end-to-end Distractor Generation framework with MoE (DG-MoE), and put forward modifications on both model structure and routing strategy for facilitating the generation quality and diversity simultaneously. We refine the standard Transformer architecture with a set of experts to improve the model's capability of identifying diverse key information, and present an unsupervised content selection layer to point out salient tokens for decoder's cross attention calculation. Besides, we propose an effective routing strategy to activate the diversity potential of MoE. Concretely, we design a minimal loss assignment mechanism for an exclusive target-to-expert allocation, and for the first time propose a stable routing strategy for diverse generation.

We conduct experiments on RACE [9]. Our proposed model greatly outperforms the previous state-of-the-art model on the generation quality of multiple distractors. Moreover, we leverage sufficient metrics to measure the diversity of different outputs, and both automatic and human evaluation validate the effectiveness of our new routing strategy for diverse generation.

To sum up, our contributions are as follows:

1. To cope with the one-to-many distractor generation task, we propose an end-to-end generative structure with MoE, which is able to produce multiple high-quality distractors and shows good interprebility.
2. We address the significance of routing strategy when applying MoE to one-to-many generation tasks, and propose two effective routing strategies for diverse generation. Experiments show they achieve great performance on both lexical diversity and semantic diversity.

2 Related Work

2.1 Distractor Generation

Distractor generation is an essential part of multiple choice question generation task. Recently, some approaches introduce hierarchical end-to-end architecture [6,11] with well-designed attention mechanism for DG task. Co-attention between passages and questions is proposed to focus on the key content in the generation process [19]. Another work [13] designs two representation reforming modules based on gate strategies to extract key information from the questions and passages. Following these work, we propose an end-to-end framework for distractor generation without manual designed features and rules.

2.2 Mixture of Experts

In recent years, Mixture of Experts (MoE) [7] is widely applied in NLP for quality enhancement [18] and inference acceleration [10]. The generative models based on MoE are always composed of independently parameterized components such as latent vectors [1], feed-forward layers [15], decoders [12,16] and attention heads in Transformer architecture [18].

MoE is also intuitively suited to generate multiple results in diverse-promoting generation tasks. A pipeline framework is proposed with a MoE based selector layer to diversify key content selection [2] . This mechanism has already been utilized in DG task [17]. In this paper, we integrate MoE into standard Transformer for diverse DG generation.

3 The Proposed Framework

We use p, q, a to denote the input passage, question and answer, respectively. MoE aims to optimize multiple experts independently in the training process

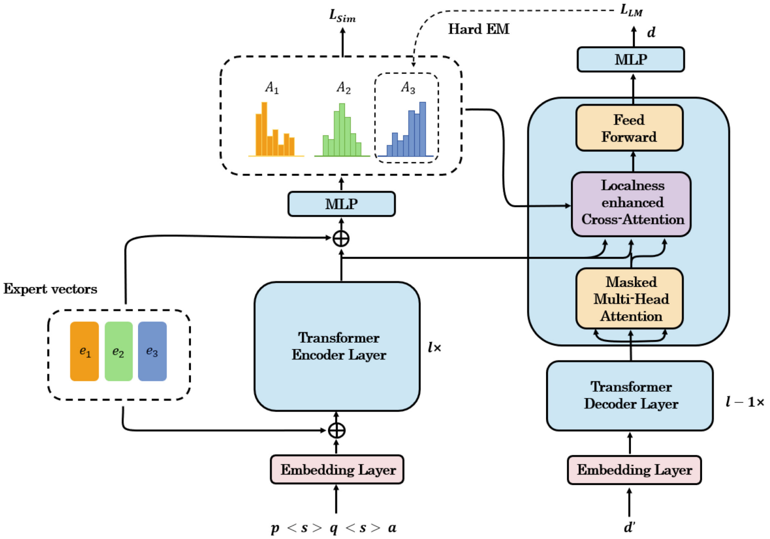


Fig. 1. The overall structure of our proposed framework of DG-MoE.

and generate diverse results with the guidance of different experts during inference. Similar to previous MoE research [2,17], we construct K random-initialized latent variables e as experts. Following the stochastic hard EM [4,8], each piece of data will be allocated to one expert by the designed routing strategy in the training process. During inference, all experts will be separately exploited to generate K different results [16].

3.1 Model Structure

We integrate the MoE module into the original encoder-decoder structure of Transformer, which is illustrated in Fig. 1.

MoE Enhanced Transformer Encoder. In the encoder, the source sequences x will be converted into token-level representations through the embedding layer. Further, an expert vector e_k will be selected out to enrich the token-level embedding and produce an expert-specific representation:

$$h_{0,k}^i = \text{Embedding}(x_i) + \lambda_e * e_k \tag{1}$$

where $h_{0,k}^i$ is the representation of the i-th token integrating the k-th expert vector and λ_e is a hyper-parameter.

A standard Transformer encoder is employed to encode the source sequence and obtain the overall representation of the p-q-a triplet:

$$H_{l,k} = \text{Encoder}(H_{0,k}) \tag{2}$$

where l is the number of encoder layers.

In order to automatically find out salient content for DG, on the top of the encoder, we apply an extra MLP layer to derive the token-level attention:

$$A_k = \sigma([H_{l,k} \oplus e_k]W_A + b_A) \tag{3}$$

$A_k \in \mathbb{R}^I$ is the attention vector associated with the k-th expert, I is the length of the input sequence, and $\oplus$ denotes vector concatenation. Intuitively, tokens with high attention scores are semantically important and should receive more focus in the distractor generation process. In the next stage, A_k will be transmitted to the decoder layer for localness-enhanced cross-attention calculation.

MoE Enhanced Transformer Decoder. To encourage the model to attach importance to the salient local semantic information when generating distractors, we design a localness-enhanced cross-attention module, which is computed as:

$$\begin{aligned}\text{Attention}(Q, K, V) &= (\lambda_d \times \text{softmax}(A_k) \\ &+ (1 - \lambda_d) \times \text{softmax}(\frac{QK^T}{\sqrt{d}}))V\end{aligned} \tag{4}$$

where Q, K are linear projections of the source representation $H_{l,k}$, and V is from the output of the self-attention module. λ_d is a hyper-parameter. Take notice that the localness-enhanced cross-attention is only applied in the last layer of the Transformer decoder.

The LM loss will be calculated as the cross entropy loss:

$$\mathcal{L}_{LM}(k) = -\frac{1}{l_d}\sum_{i=0}^{l_d} y_i \text{log} p_{i,k} \tag{5}$$

where $\mathcal{L}_{LM}(k)$ is the LM loss with the k-th expert, and l_d is the length of the reference distractor.

3.2 Routing Strategy

The routing strategy determines the data-to-expert allocating relationship in MoE training. A proper routing strategy is able to activate the diversity potential of the MoE model and is the key to a successful training [16]. Therefore, we propose a novel routing strategy based on the stochastic hard EM method, which is specially designed for one-to-many generation tasks.

Stochastic Hard EM. The hard EM routing strategy [4,8] is widely utilized along with MoE method in various fields, which consists of E and M two steps.

E step: Given a piece of data, forward the generation process with every expert vector without dropout, and choose the best expert $e_{k'}$ with the minimal LM loss:

$$k' = \arg\min_{k} \mathcal{L}_{LM}(k) \tag{6}$$

M step: Update the parameters with the chosen expert vector $e_{k'}$.

Hard EM with Minimal Loss Assignment. The hard EM method realizes the specialization of experts. Accordingly, a particular expert will be responsible for one training sample, and the experts are assigned independently for different samples. As a result, when multiple target distractors with respect to one source sequence are given, they are possibly allocated to the same expert with the hard EM method.

Ideally, different distractors from one source sequence should be allocated to different experts. In this way, the model encourages different experts to focus on distinctive localness information with the same input during inference. To this end, we modify the original hard EM strategy as follows:

E step: Given n reference distractors with respect to one source sequence, forward the generation process for each output distractor with every expert latent vector without dropout. Gather all the LM loss to a matrix $L \in \mathbb{R}^{n*K}$, where K is the number of experts and $K \geq n$. Based on the loss matrix L, we carry out the minimal weight assignment, which assigns n training samples to n different experts with the minimum sum of LM loss.

M step: Update the parameters with the chosen expert set from E step.

We adopt the Scipy toolkit[1] to implement the minimal weight assignment between the experts and training samples.

Stable Routing Strategy. For the EM-based routing strategy, the target expert for the same input may frequently change in different training epochs, which is called the unstable problem in MoE [14]. A stable routing strategy has been proved to be efficient for improving the generation quality [3], it is also vital for generating diverse outputs according to our observation.

In this work, we propose a simple approach to reduce the fluctuation of the routing process. We divide the training process into three steps: 1) The first stage follows the learn-to-route paradigm, and trains the MoE model by the hard EM with minimal loss assignment for a short period of time. This roughly trained model will serve as a router. 2) Forward the router on the training set without back propagation, and record the data-to-expert allocating relationship as the fixed assignment policy. 3) Finetune the DG model from scratch and allocate each training sample to experts with the fixed assignment policy.

3.3 Loss Function

In the MoE enhanced encoder, we present an extra layer on top of the Transformer encoder to calculate a token-level attention vector A_k. To further encour-

[1] https://scipy.org/.

age each expert to focus on different parts of the source sequence, we propose a diverse loss as follow:

$$\mathcal{L}_{Sim} = \frac{1}{n(n-1)} \sum_{i \in E} \sum_{j \in E}^{j \neq i} \text{sim}(A_i, A_j) \tag{7}$$

where E is the set of selected experts for n distractors, $\text{sim}(\cdot)$ is the cosine similarity. Since n distractors should be allocated to n different experts in Eq. (7), the diverse loss will only be appended when the minimal loss assignment routing strategy is applied.

The final loss is calculated as:

$$\mathcal{L} = \mathcal{L}_{LM} + \gamma \mathcal{L}_{Sim} \tag{8}$$

where γ is the weight parameter.

4 Experimental Setup

4.1 Dataset

We conduct extensive experiments on RACE [9], which is collected from the English exams of middle and high schools in China. Previous work [6] has filtered out the distractors that are not semantically related with the passage and removed the fill-in-the-blank questions. The detailed statistics of the cleaned RACE dataset is shown in Table 2.

Table 2. The statistics of RACE-Gao dataset, d, p, q, a refer to distractor, passage, question, answer respectively.

	# d	# q-a pair	# p	Avg. d per q-a pair	Avg. q-a pairs per p
Train	96382	44834	20157	2.15	2.22
Valid	12079	5717	2521	2.11	2.27
Test	12273	5771	2519	2.13	2.29

4.2 Comparing Methods

Hierarchical Static Attention (HSA) [6]: A hierarchical generation model with static attention that limit the correlation of answers and distractors.

Hierarchical Co-Attention (HCA) [19]: A hierarchical generation model with co-attention between passage and question.

EDGE [13]: An end-to-end generation framework with semantic enriching module and representation reforming module.

Hierarchical Multi-Decoder Network (HMD-Net) [12]: A multi-decoder framework with an attention deviation method.

MSG-Net [17]: A pipeline generation framework with MoE method which extracts key sentences as the guidance of generation.

Bart + Top-k/Nucleus sample [5]: A Bart-base model with Top-k/Nucleus sampling decoding method.

Bart + Jaccard Distance: A Bart-base model that utilizes a Jaccard distance threshold of 0.5 to select three diverse distractors, and we follow the hyper-parameter settings of [19].

4.3 Automatic Evaluation Metrics

We apply the following metrics for automatic evaluation. **BLEU, ROUGE**: We apply corpus-level BLEU-4, ROUGE-1 and ROUGE-L to measure the quality of the generated results. **BertScore**: BertScore is a supplement to accuracy based metrics and is used to measure the semantic similarity between the references and generated results. **Pairwise BLEU, Distinct**: We adopt Pairwise BLEU, Distinct-1 and Distinct-2 to measure the generation diversity.

Table 3. Experimental results on RACE dataset with 3 distractors comparing with baselines. B4 refers to BLEU-4, R-1, R-L refer to Rouge-1, Rouge-L, BS refers to BertScore.

Models	1-st B4	2-nd B4	3-rd B4	Avg B4	Avg R-1	Avg R-L	Avg BS
HSA [6]	6.43	5.17	4.59	5.40	15.63	14.67	-
HCA [19]	7.01	5.51	4.88	5.80	16.06	15.12	-
EDGE [13]	7.57	6.27	5.70	6.51	18.66	18.27	-
HMD-Net [12]	7.66	6.37	5.33	6.45	-	24.99	-
MSG-Net [17]	8.87	8.86	8.53	8.75	-	26.39	-
Bart+Top-k	8.07	7.96	8.49	8.17	25.14	24.55	89.59
Bart+Nucleus	8.34	8.38	8.21	8.31	26.35	25.37	89.66
Bart+Jaccard	10.45	9.02	8.32	9.26	27.13	26.45	89.74
Ours	9.52	9.12	9.59	**9.41**	**27.46**	**26.80**	**89.78**

4.4 Implementation Details

We build our model based on Bart-base with HuggingFace Transformers library. We adopt Adam optimizer and set the maximum learning rate, batch size, and warmup step to 10^{-5}, 20, 3000. When applying stable routing strategy, we train the router for 3 epochs in the first stage. Then we finetune the overall model with the fixed routing policy for 6 epochs from scratch. We set the expert number K as 3, λ_e and λ_d as 0.1, and γ as 5.

5 Results and Analysis

5.1 Main Results

The experimental results for generation accuracy on RACE dataset are shown in Table 3. Our proposed model brings an obvious performance gain on the generation quality, achieving a 7.5% improvement on the average BLEU-4 compared with the previous SOTA model MSG-Net. Another noteworthy fact is that the traditional methods based on Jaccard Distance will incur the problem of quality decline when the generated number increase, while our model maintains a high level for all three different distractors (above 9.00 BLEU-4).

Table 4. Ablation study. BS refers to BertScore, PB refers to Pairwise BLEU, Dis refers to Distinct. †: Since the diverse loss will only be appended when minimal loss assignment routing strategy is applied, it will be removed at the same time.

Model Variants	Avg B4	Avg BS	PB ⇓	Dis-1	Dis-2
Our full model	9.41	89.78	19.36	69.61	82.40
- Localness-enhanced cross-attention	9.43	89.77	21.81	67.13	80.44
- Diverse loss	9.57	89.82	24.77	67.40	78.61
- Stable routing strategy	9.17	89.65	21.35	69.83	81.69
- Minimal loss assignment†	9.36	89.68	37.26	59.10	68.83

5.2 Ablation Studies

In this section, we will investigate how different model structures and routing strategies influence the final performance of distractor generation. The results are shown in Table 4.

For the influence of the different model structures, the localness-enhanced cross-attention and diverse loss method both strengthen the generation diversity with the loss of the generation quality. Our proposed methods get an appropriate balance between these two aspects.

On the other side, we can clearly observe that both generation quality and diversity metrics decrease when removing our proposed stable routing strategy. On this basis, it leads to an even sharper decrease on generation diversity when we remove the minimal assignment method.

As the stochastic hard EM may allocate different targets from one source to the same expert, it will degenerate from target-to-expert to source-to-expert paradigm. Moreover, the minimal assignment method with diverse loss can enhance the semantic diversity by making experts focus on different local information under the same input.

Interestingly, the generation diversity on the same structure performs much differently with various strategies. Combining the results in Table 3 and Table 4, we could infer that the model structure provides the model with the upper bound of diversity rather than the final capacity. In the meanwhile, it's up to the routing strategy that how much this upper bound can be brought into play.

5.3 Analysis on the Number of Experts

Our model is able to produce a pre-defined number of distractors with different K. We perform experiments to verify the influence of K value on the results.

Table 5. The result in different number of generated distractors.

K	Avg B4	Avg BS	PB ⇓
3	9.41	89.78	19.36
4	9.34	89.77	18.69
5	8.92	89.56	17.53

When we apply the stable routing strategy in the training process, the number of valid training steps for each expert declines while the number of the experts grows. As a result, when generating more outputs, the quality decreases while the generation diversity increases, as shown in Table 5.

5.4 Attention Analysis

Table 6 illustrates one case about how the attention vector influences the performance of our model. One of the generated distractors is quite similar to the given answer in the sentence structure with totally different keyword. The other two distractors are both fluent and semantic-related with the passage. What's more, the three generated distractors show high semantic diversity.

According to the attention heatmap of Table 6, there are two factors that make our attention mechanism capable of improving the generation diversity. First, there exists a clear boundary between the attention map of different experts. Second, the generated distractors are highly semantic-consistent with the focused contents.

5.5 Human Evaluation

We conduct human evaluation to compare our proposed model with HSA [6] and Bart-base with Jaccard Distance method. We randomly selected 100 samples from RACE and asked three annotators to score the generated results in the scale of [1, 5], from the following 5 aspects. **Fluency**: whether the generated distractor is grammatical and fluent. **Relevancy**: the semantic relevancy between the generated distractor and the input *p*-*q*-*a* triplet. **Distracting**: the capability of the generated result to confuse readers from identifying the correct answer.

Table 6. Attention analysis when expert number is 3. Tokens in red, green and blue respectively refer to the focused tokens with three different experts and the color brightness reflects the actual attention score.

Input Question: according to barack obama, the highest praise for steve jobs is that

Input Answer: people learned about his death from his products

Reference Distractors:

his products are widely used all over the world
his products are thought to be beautiful and easy to use
people all over the world are sad at his death

Passage

have you listened to music on an ipod ? ... jobs , one founder of apple computers , died on wedsday , october 5 , 2011 at the age of 56 . ... the macintosh , the ipod , the iphone , the ipad . people found his products beautiful and easy to use . ... by making computers personal and putting the internet in our pockets , he made the information revolution not only easy to reach , but fun ... by creating products that surprised people , he changed the way people worked and lived their everyday lives .

Generated Result with Expert 1: people learned about his life from his products

Generated Result with Expert 2: he made computers personal and put the internet in our pockets

Generated Result with Expert 3: his products are beautiful and easy to use

Table 7. The human evaluation results of generated distractors on RACE.

	Fluency	Relevancy	Distracting	Diversity	Difference
HSA [6]	4.45	3.91	3.19	4.24	4.42
Bart+Jaccard Distance	**4.88**	4.62	3.48	4.08	4.20
Our DG-MoE	4.83	**4.66**	**3.59**	**4.26**	**4.49**
Spearman	0.255	0.152	0.441	0.594	0.557

Diversity: the lexical diversity of the multiple generated results. **Difference**: the semantic difference between different generated results.

The results are shown in Table 7. HSA shows obvious disadvantage in generation quality (in terms of fluency, relevancy and distracting). Jaccard Distance method greatly underperforms on lexical diversity semantic difference. Our proposed model achieves the highest scores on all metrics except fluency.

6 Conclusion

In this paper, we propose a novel generation framework for distractor generation (DG-MoE). We enhance the Transformer architecture with MoE to generate multiple diverse hypotheses. Besides, we explore the influence of different routing strategies when applying MoE method on one-to-many generation task, and propose two novel approaches based on the hard EM method. Experimental results show that our model achieves great performance on both generation quality and diversity, and outperforms existing models on both automatic and human evaluation metrics. Our proposed MoE model enriched with the effective routing strategy can be applied to other text generation tasks.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (62076008) and the Key Project of Natural Science Foundation of China (61936012).

References

1. Bi, S., et al.: Simple or complex? Complexity-controllable question generation with soft templates and deep mixture of experts model. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021, pp. 4645–4654. Association for Computational Linguistics (2021)
2. Cho, J., Seo, M.J., Hajishirzi, H.: Mixture content selection for diverse sequence generation. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 3119–3129. Association for Computational Linguistics (2019)
3. Dai, D., et al.: StableMoE: stable routing strategy for mixture of experts. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022, pp. 7085–7095. Association for Computational Linguistics (2022)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc.: Ser. B (Methodol.) **39**(1), 1–22 (1977)
5. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 889–898. Association for Computational Linguistics, Melbourne, Australia, July 2018
6. Gao, Y., Bing, L., Li, P., King, I., Lyu, M.R.: Generating distractors for reading comprehension questions from real examinations. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 6423–6430. AAAI Press (2019)
7. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991)
8. Kearns, M.J., Mansour, Y., Ng, A.Y.: An information-theoretic analysis of hard and soft assignment methods for clustering. In: Jordan, M.I. (ed.) Learning in Graphical Models, NATO ASI Series, vol. 89, pp. 495–520. Springer, Netherlands (1998). https://doi.org/10.1007/978-94-011-5014-9_18
9. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: RACE: large-scale ReAding comprehension dataset from examinations. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 785–794. Association for Computational Linguistics, Copenhagen, Denmark, September 2017

10. Lepikhin, D., et al.: GShard: scaling giant models with conditional computation and automatic sharding. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021
11. Li, J., Luong, M., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Beijing, China, 26–31 July 2015, Volume 1: Long Papers, pp. 1106–1115. The Association for Computer Linguistics (2015)
12. Maurya, K.K., Desarkar, M.S.: Learning to distract: a hierarchical multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 19–23 October 2020, pp. 1115–1124. ACM (2020)
13. Qiu, Z., Wu, X., Fan, W.: Automatic distractor generation for multiple choice questions in standard tests. In: Scott, D., Bel, N., Zong, C. (eds.) Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), 8–13 December 2020, pp. 2096–2106. International Committee on Computational Linguistics (2020)
14. Roller, S., Sukhbaatar, S., Szlam, A., Weston, J.: Hash layers for large sparse models. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, 6–14 December 2021, Virtual, pp. 17555–17566 (2021)
15. Shazeer, N., et al.: Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings (2017)
16. Shen, T., Ott, M., Auli, M., Ranzato, M.: Mixture models for diverse machine translation: tricks of the trade. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 5719–5728 (2019)
17. Xie, J., Peng, N., Cai, Y., Wang, T., Huang, Q.: Diverse distractor generation for constructing high-quality multiple choice questions. IEEE ACM Trans. Audio Speech Lang. Process. **30**, 280–291 (2022)
18. Zhang, S., Feng, Y.: Universal simultaneous machine translation with mixture-of-experts wait-K policy. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7–11 November 2021, pp. 7306–7317. Association for Computational Linguistics (2021)
19. Zhou, X., Luo, S., Wu, Y.: Co-attention hierarchical network: generating coherent long distractors for reading comprehension. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020, pp. 9725–9732 (2020)

Improve the Diversity and Novelty for Open-Ended Neural Text Generation via Inverse Probability Weighting

Xinran Zhang[1,2,3](✉), Maosong Sun[1,2,3,4], Jiafeng Liu[1,2,3], and Xiaobing Li[1,2,3]

[1] Department of AI Music and Music Information Technology, Central Conservatory of Music, Beijing, China
zhangxr.wspn@gmail.com

[2] Key Laboratory of Music and Brain Science, Central Conservatory of Music, Ministry of Education, Beijing, China

[3] Laboratory of AI Music, Central Conservatory of Music, Laboratory of Philosophy and Social Sciences, Ministry of Education, Beijing, China

[4] Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract. Stochastic sampling methods for open-ended neural text generation severely affect the diversity and novelty of generated texts. Traditional sampling methods only prune the low-likelihood part of the predicted distribution to trade-off between the fluency and diversity of generated texts. They do not manipulate the high-likelihood part, which leads to the likelihood trap that induces low diversity and boredom. They also do not directly leverage that human does not always favor high-likelihood texts. Inspired by these, we propose a novel sampling method that rescales the high-likelihood part of the distribution with inverse probability weighting. It increases the diversity and novelty of generated texts by rescaling and penalizing the high-likelihood part of the predicted distribution, and preserves the fluency by using multi-filtering truncation on the low-likelihood part. Experimental results show that compared with traditional methods our algorithm can significantly increase the diversity and novelty of generated texts without sacrificing fluency.

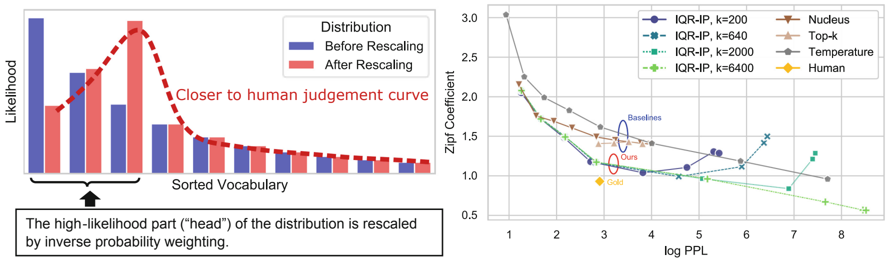
Keywords: Text degeneration · Quality-diversity trade-off · Diversity and novelty

1 Introduction

Open-ended neural text generation is greatly affected by decoding methods. Counter-intuitively, traditional quality-oriented decoding methods such as beam search, which maximizes the likelihood of decoded texts, induces the well-known *text degeneration* [7,14] and *likelihood trap* [1,16]. As a result, many works have focused on stochastic sampling method such as top-k sampling [4,8] or nucleus sampling (top-p sampling, [7]). These methods prune the low-likelihood part

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 774–785, 2023.
https://doi.org/10.1007/978-3-031-44693-1_60

of the language model's predicted distribution to trade-off between fluency and diversity. Recent works [3,11,16] reveal that these methods achieve on-par performance regarding their quality-diversity trade-off feature, i.e., improving the same amount of diversity costs equal amount of fluency. Still, there exist undiscovered properties to understand better the relationship between stochastic sampling algorithms and open-ended neural text generation [11].



(a) Illustration of our algorithm. (b) Comparison of the trade-off curve.

Fig. 1. (a) The high-likelihood part of the language model's predicted distribution on each sampling step is rescaled by inverse probability weighting to penalize the high-likelihood words. (b) The achieved trade-off curve of our method is considerably closer to the human-level point than all baseline methods.

We note that none of the traditional sampling algorithms have directly manipulated the high-likelihood part of the distribution since high-likelihood words are always considered to be "trustworthy". Essentially, the observed quality-likelihood curve by human judgment is inversely proportional to the likelihood in the high-likelihood area [16], which confirms the intuition that human does *not* always favor high-likelihood words [7,14]. Inspired by these, we propose the *interquartile range inverse probability* (IQR-IP) sampling algorithm. It increases the diversity by rescaling and penalizing the high-likelihood part of the predicted distribution with inverse probability weighting, and preserves the fluency by using multi-filtering truncation on the low-likelihood. The rescaled distribution will achieve a closer resemblance to the quality-likelihood curve (such as the human judgment of Fig. 1 by [16]), as is illustrated in Fig. 1a. Empirical results show that our algorithm achieves significantly better trade-off behavior than baseline methods (illustrated in Fig. 1b). Human evaluation results also suggest that our method can increase the novelty of generated text without corrupting the fluency.

2 The Likelihood Trap

2.1 Trapped Trajectory Induced by the High-Likelihood

We study the likelihood trap and the low-diversity/novelty issue in open-ended generation cases. Since the sampling process is stochastic, it is necessary to generate a sufficiently large number of samples for a concerned input prompt to

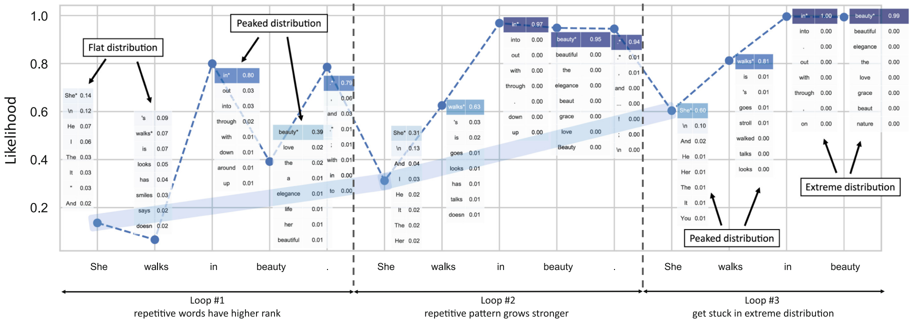


Fig. 2. The trajectory of predicted likelihood ("o" marker) and predicted distribution (heatmap box beside each marker in "word-likelihood" format, with the sampled word marked by "*") for the first three repetition loops.

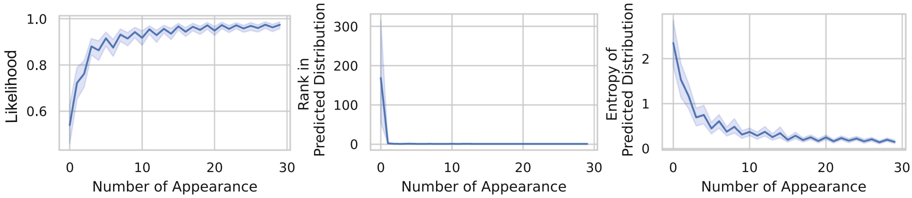


Fig. 3. Trajectories of repetitive words extracted from samples that contain repetition loops. Repetitive words that appear more than 30 times are extracted and aligned to form their trajectories.

obtain statistically significant results. In order to create a feasible setting, we simply choose a fixed prompt as "She walks in beauty" (from Lord Byron's poetry) to set a high-novelty reference, and generate 5,000 continuations for 200 tokens to observe the generation trajectory. We use GPT-2 Small [12] with nucleus sampling ($p = 0.95$). To detect trapped repetitions on the generated passages, we adopt the n-gram entropy metric [5,13,17] by calculating the entropy of n-gram distribution in a fixed-length token window. Empirically, we found that the entropy threshold of 2.0 for unigram on 200-length token windows is good enough to filter repetition. We present a generated continuation that contains *infinite loops* of the prompt, in which case the generation process gets "stuck" in repeating the input prompt. The likelihood trajectory of first 3 loops is presented in Fig. 2. We report the following observations.

- Repetitive words always have *high likelihood* and *high rank* in the predicted distribution (see "*" labeled words in each heatmap box in Fig. 2).
- Repetition tendency *grows stronger* when *more loops occur* (due to a few sampling steps that happen to pick repetitive token in non-extreme distribution, e.g., in Loop #2), as the *flat distribution* in Loop #1 (e.g., "She" and "walks") gradually becomes *peaked distribution* in Loop #3, and peaked distribution in Loop #1 (e.g., "in" and "beauty") becomes *extreme distribution*

in Loop #3, which reciprocally contributes to stronger repetition pattern in the context.
- The predicted distribution got *stuck* in *extreme distribution* that assigns almost all probability mass for repetitive words (e.g., "in" and "beauty" in Loop #3).

To further verify these phenomena, we extract and align the trajectories of each repetitive word that occurs more than 30 times in the context from all generated passages to observe its overall trajectory. Figure 3 presents the trajectories of likelihood, rank in predicted distribution, and entropy of predicted distribution, where x axis is the number of the appearance of repetitive words. After a few appearances of repetitive words, the predicted distribution will quickly get stuck in extreme distribution where predicted probability approaches 1, rank approaches 1, and entropy approaches 0, rendering infinite repetition loops. The undesired behavior of high-likelihood words on the predicted distribution induces the likelihood trap and leads the model to exhibit repetition behavior.

2.2 Improving Diversity by Penalizing the High-Likelihood

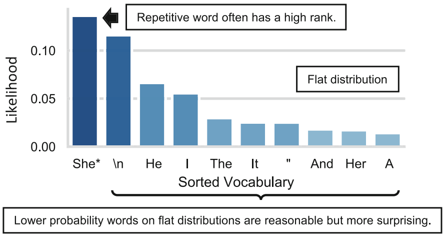


Fig. 4. Illustration of the high-likelihood on the flat distribution.

We present a detailed observation of the high-likelihood words in Fig. 4, which shows the first sampling step of Loop #1 from Fig. 2 (the context was "She walks in beauty."). Besides "She" that has the highest predicted likelihood, lower-likelihood words ("\n", "He", "I", "The", ...) are also reasonable. If we rescale the distribution and emphasize these lower-likelihood words to improve the diversity and novelty, the fluency of generated passage will not be compromised. Besides, it is proven beneficial to increase generation diversity by emphasizing less probable words during training [14]. Furthermore, human judgment exhibits an inverse correlation to the likelihood in the high-likelihood part (Fig. 1, [16]). Inspired by these, we adopt the *inverse probability weighting* method that is commonly seen in causal inference (see Chapter 2, [6]). We first identify a small subset of high-likelihood words that contains all reasonable choices (such as in Fig. 4). Then adopt inverse probability weighting to rescale the distribution of the "head" and penalize the high-likelihood, as is illustrated in Fig. 1a.

3 Interquartile Range Inverse Probability Sampling Algorithm

3.1 Fine-Grained Filtering on the Low-Likelihood

The primary difficulty in identifying the high-likelihood "head" to rescale is the variation of the shape of the predicted distribution, i.e., the discrepancy between the flat distribution and the peaked distribution [7]. Intuitively, the *interquartile range* (IQR) can adapt to such variation since it is based on quantile. Furthermore, we also need to leverage the traditional filtering methods, which truncate low-likelihood words to preserve fluency and ensure that reliable words are kept to calculate IQR. As a result, we propose to perform fine-grained filtering on the low likelihood.

Let $p_{LM}(x_t|x_{1:t-1})$ denote the auto-regressive language model's predicted probability of word x_t from vocabulary V given its context $x_{1:t-1}$ on time step t [2]. All the following manipulations are conducted across all possible t. For simplicity, we directly use $p(x)$ to represent $p_{LM}(x_t|x_{1:t-1})$. We propose to jointly filter an initial subset V_{fil} out of V using top-k filtering (with parameter k) and nucleus filtering (with parameter p).

$$V_{fil} = \text{top-}k(V) \cap \text{nucleus-}p(V). \tag{1}$$

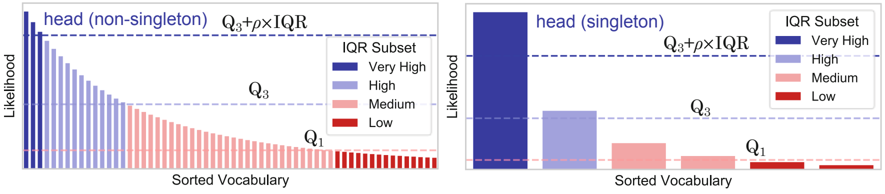
Let $p_{fil}(x)$ denote the normalized distribution on V_{fil}. We propose to calculate IQR of $p_{fil}(x)$, that is, calculate 75% percentile of $p_{fil}(x)$ as Q_3, 25% percentile as Q_1, let $IQR = Q_3 - Q_1$ (all scalar), and divide V_{fil} into subsets by using likelihood threshold determined by IQR as follows.

$$\begin{matrix} V^{VeryHigh} & : p_{fil}(x) \geq Q_3 + \rho \times IQR \\ V^{High} & : Q_3 + \rho \times IQR > p_{fil}(x) \geq Q_3 \\ V^{Medium} & : Q_3 > p_{fil}(x) \geq Q_1 \\ V^{Low} & : Q_1 > p_{fil}(x) \geq Q_1 - \rho \times IQR \end{matrix}, \tag{2}$$

$$p_{inv}(x) = \begin{cases} \left(\sum_{x \in V^{VeryHigh}} p'_{fil}(x)\right) \times \frac{p'_{fil}(x)^{-1}}{\sum_{x \in V^{VeryHigh}} p'_{fil}(x)^{-1}}, \forall x \in V^{VeryHigh} \\ p'_{fil}(x), \qquad otherwise \end{cases}. \tag{3}$$

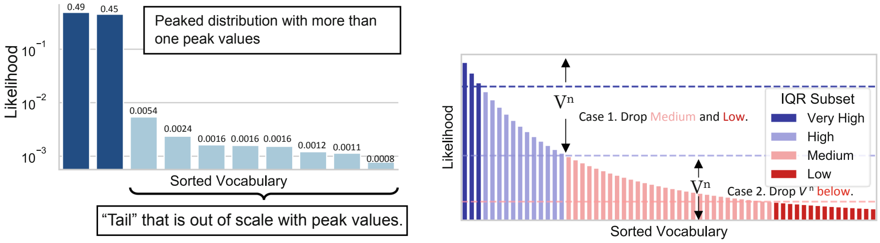
where ρ is the hyperparameter for IQR coefficient with the typical value being 1.5. The division is illustrated in Fig. 5. Considering the outlier-identifying nature of IQR, $V^{VeryHigh}$ can be regarded as the "head" part that we need to rescale, which we expect that the likelihood of the least probable word in $V^{VeryHigh}$ is still "high enough" to be reasonable choices (Fig. 5a). Since IQR is based on the quantile, $V^{VeryHigh}$ will be singleton on peaked distribution with low entropy (Fig. 5b). In that case, manipulating and redistributing the probability mass of $V^{VeryHigh}$ does not have any effect. It will not corrupt peaked distribution cases with "unquestionably right" words.

We also consider a particular case of distribution. Figure 6a presents an example of peaked distribution with more than one peak value. A small value of p for nucleus sampling will miss the second peak, while a large value of p will let in low-likelihood words that are out of scale with peak values. We note that it can be resolved by considering the scale constraint of likelihood. Concretely, we propose a novel filtering method by defining a scale threshold as the fraction of



(a) Flat distribution case. (b) Peaked distribution case.

Fig. 5. Illustration of IQR subset division.



(a) Multi-peak distribution. (b) Adaptive pruning from Equation 5

Fig. 6. Illustration of Top1CTRL filtering.

the maximum likelihood of the predicted distribution. We name it as the "Top-1 Controlled" (Top1CTRL) filtering with parameter n as follows.

$$V^n = \{x \mid p(x) \geq \max p(x)/n, x \in V\}. \tag{4}$$

$$V'_{fil} = \begin{cases} V^{VeryHigh} \cup V^{High}, & if \quad V^n \subseteq (V^{VeryHigh} \cup V^{High}) \quad \text{(Case 1)} \\ V_{fil} \cap V^n, & otherwise \qquad \text{(Case 2)} \end{cases}. \tag{5}$$

Note that a small value of n might over-prune the vocabulary and harm the diversity. As a result, we propose to use V^n to prune V_{fil} in a fine-grained manner, as is described in Eq. 5 and Fig. 6b. Case 1 ensures that V^n does not over-prune words categorized as "Very High" or "High" since they are identified by IQR and are likely to contain reasonable words. Case 2 describes other cases where V^n directly truncates V_{fil} and works jointly with nucleus filtering and top-k filtering. The pruned set is denoted by V'_{fil}. Empirically, n can be set to a fixed value of 100 to achieve good performance.

3.2 Inverse Probability Weighting on the High-Likelihood

With V'_{fil} acquired, we propose to redistribute the probability mass for each word in $V^{VeryHigh}$ (i.e., the "head") proportionally to its inverse probability while keeping the sum of probability mass in $V^{VeryHigh}$ constant. Let $p'_{fil}(x)$ denote the normalized distribution on V'_{fil}. The transformation on $V^{VeryHigh}$ is described in Eq. 3 and Fig. 1a, where $p_{inv}(x)$ denotes the rescaled distribution.

Finally, the sampling is performed with $p_{inv}(x)$. We refer to the above algorithm as the *interquartile range inverse probability* (IQR-IP) sampling algorithm. The main features of our algorithm are as follows.

A. We use fine-grained truncation on low-likelihood "tail" with 3 parameters (p, k, and n). It aims to control the "tails" to preserve fluency and guarantee the correct identification of the "head". Empirically, these parameters can be fixed around the reference point to achieve good performance.

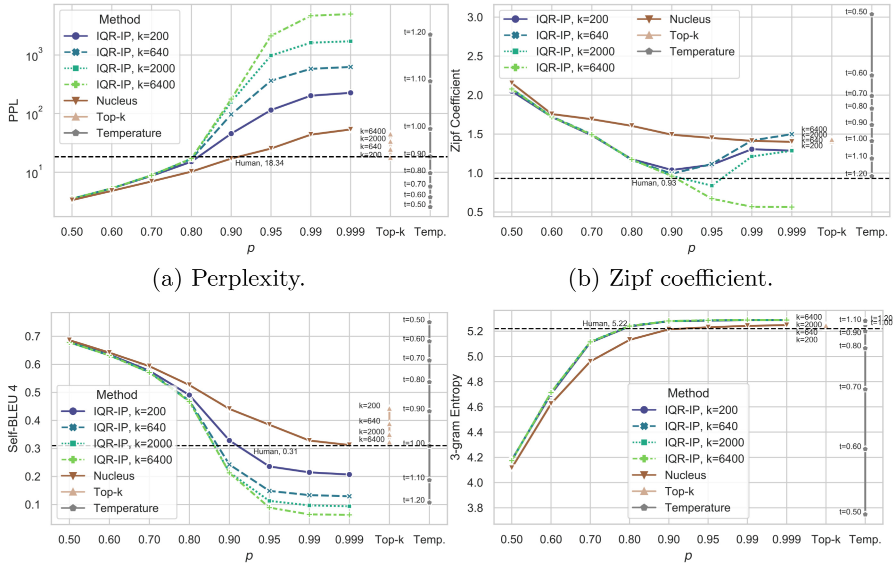
B. The distribution of the high-likelihood "head" identified by IQR is rescaled by inverse probability weighting using Eq. 3. It aims to improve diversity by penalizing the high-likelihood words, resembling the quality-likelihood curve of human judgment.

4 Empirical Results

The primary characteristic of our algorithm is to break the monotonicity of the predicted distribution. This is a very risky manipulation, since filtering methods that violate the *entropy reduction property* or *slope preservation property* will result in drastic performance degradation on the quality-diversity plane [11], i.e., despite huge diversity gain, the fluency might be dramatically decreased and sacrificed. As a result, we seek to compare with the monotonicity-keeping baseline method regarding the quality-diversity trade-off feature to unearth the property of our method. We choose the most widely adopted golden methods, i.e., the nucleus sampling, top-k sampling, and temperature sampling as baseline methods, which all preserve the monotonicity of the predicted distribution. For generation, we use the pre-trained GPT-2 XL model released by [15] (without any fine-tuning). To guarantee reproducibility, we adopt the settings in Sect. 2.1 to use a single and sharing prompt with a high novelty reference, and generate 5,000 continuations per hyperparameter per sampling method for the prompt. We consider the following automatic metrics.

Fluency. We calculate the averaged perplexity (PPL) of the generated texts [1,7,9] to reflect fluency. Note that the metric does not equal quality since low-perplexity texts might be repetitive and boring, while high-perplexity texts might be unreasonable. Like most existing works, we compare the metric w.r.t the human-level metric.

Diversity. We calculate the Zipf coefficient [19], a linguistic feature that reflects the sloping tendency of word frequency distribution on a corpus. We also calculate the Self-BLEU (4 and 5) score [18] that reflects the overlapping between different generated samples. We then calculate n-gram entropy [17] that reflects the diversity of n-gram distribution and repetition tendency.



(a) Perplexity.

(b) Zipf coefficient.

(c) Self-BLEU 4.

(d) 3-gram entropy.

Fig. 7. Empirical results for the generated texts. The human-level Zipf coefficient and self-BLEU scores are from [7]. The human-level 3-gram entropy are computed on the training dataset of WikiText-103 by [10].

The main results are shown in Fig. 1b. Clearly, although our algorithm *does* violate all three properties by [11], it counter-intuitively achieves better quality-diversity trade-off performance. The trade-off curve of our method is considerably closer to the human-level point than all baseline methods. To provide more details, we present the metric variation by tuning hyperparameters in Fig. 7. Results for the fluency metric show that our algorithm achieves human-level PPL with *more strictly filtered vocabulary.* As is shown in Fig. 7b, our algorithm can fit identical Zipf coefficient to human-level metric, while traditional sampling methods can't. It indicates that the rescaling transformation of our algorithm renders flatter and less concentrated distribution of words, which is closer to the human-level metric and unable to achieve by traditional sampling methods. Furthermore, the self-BLEU scores achieved by our algorithm decrease significantly faster than traditional methods, which indicates great diversity gain. Note that it can achieve almost the same score with "pure sampling" (near nucleus sampling with $p = 0.999$, temperature sampling with $t = 1.0$), representing the upper bound of diversity for traditional methods. It suggests that the diversity boundary of traditional methods is limited, while our method effectively expands the diversity boundary. Similarly, results for 3-gram entropy show that the entropy metric of our algorithm grows faster and achieves the human-level metric with less "tail". These results reveal that our algorithm achieves human-level diversity metrics by truncating more "tails" than traditional methods and compensating the diversity loss by rescaling the high-likelihood, which results in higher diversity than traditional methods.

5 Human Evaluation

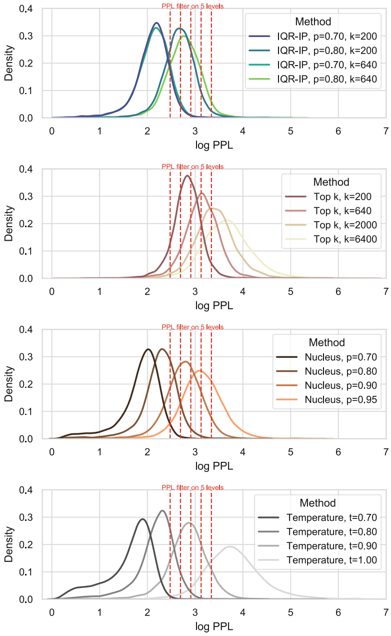


Fig. 8. Illustration of the quality variation of generated texts.

It is noteworthy that the quality of generated texts can be highly variable. With a fixed sampling parameter, the generated texts yield a distribution of PPL with variational fluency, as is shown in Fig. 8. Directly picking and evaluating these texts could be unfair, since one sampling method per parameter could have picked a bad result with low fluency, while another one with high fluency. Also, the distributional feature demands a large number of samples that sufficiently cover and represent the distribution to achieve meaningful results, which requires an extremely high monetary cost. To address these issues, we propose an on-equal-footing fluency evaluation paradigm. To cancel the impact of quality variation, we propose to use pre-defined PPL filters (colored in red vertical lines in Fig. 8) to collect generated texts with PPL around these filters (PPL ±0.5 on each level) for all sampling parameters in a post-decoding manner. In that case, the collected texts can be viewed to be equally fluent on each level for human evaluation, regardless of their sampling parameters.

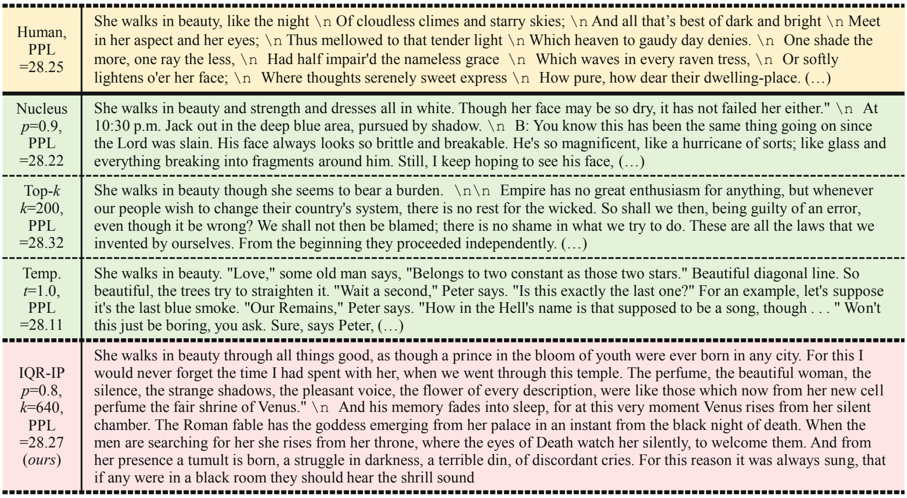
We set five pre-defined targets of PPL and collect the generated texts near these targets from all hyperparameter configurations for all sampling algorithms in a post-decoding manner. In that case, we can collect an equal number of texts per PPL level per method, measured to be similarly fluent on each level without traversing the distribution of PPL to perform a fair comparison. We then decompose the quality (overall) metric into the fluency metric and the novelty metric for human evaluation since high-likelihood texts with low quality are expected to be still fluent but boring. Following common paradigm [9,11,16], we use Amazon Mechanical Turk (MTurk) for human evaluation.

Results are shown in Table 1, which indicates that our algorithm achieves both higher fluency and novelty than traditional methods. We present generated samples with PPL near the reference text in Table 2. Under the same PPL

Table 1. Automatic metrics and corresponding human evaluation results. The metrics are averaged on generated texts from all fluency levels in Fig. 8 per sampling method. Abbreviations include self-BLEU 4/5 (SB-4/5), Zipf coefficient (ZC), 3-gram entropy (Ent-3), fluency (F), novelty (N) and overall (O).

Method	SB-4 ↓	SB-5 ↓	ZC ↓	Ent-3 ↑	F ↑	N ↑	O ↑
Nucleus	0.31	0.18	1.49	5.24	3.70	3.59	3.65
Top-*k*	**0.30**	**0.17**	1.39	5.24	3.79	3.49	3.64
Temperature	0.32	0.19	1.60	5.23	3.74	3.53	3.64
IQR-IP (*ours*)	0.31	0.18	**1.19**	**5.25**	**3.83**	**3.67**	**3.75**

Table 2. Generated examples that are on the same fluency level (with PPL near the reference passage, and can be viewed as similarly fluent).



Human, PPL =28.25	She walks in beauty, like the night \n Of cloudless climes and starry skies; \n And all that's best of dark and bright \n Meet in her aspect and her eyes; \n Thus mellowed to that tender light \n Which heaven to gaudy day denies. \n One shade the more, one ray the less, \n Had half impair'd the nameless grace \n Which waves in every raven tress, \n Or softly lightens o'er her face; \n Where thoughts serenely sweet express \n How pure, how dear their dwelling-place. (…)
Nucleus *p*=0.9, PPL =28.22	She walks in beauty and strength and dresses all in white. Though her face may be so dry, it has not failed her either." \n At 10:30 p.m. Jack out in the deep blue area, pursued by shadow. \n B: You know this has been the same thing going on since the Lord was slain. His face always looks so brittle and breakable. He's so magnificent, like a hurricane of sorts; like glass and everything breaking into fragments around him. Still, I keep hoping to see his face, (…)
Top-*k* *k*=200, PPL =28.32	She walks in beauty though she seems to bear a burden. \n\n Empire has no great enthusiasm for anything, but whenever our people wish to change their country's system, there is no rest for the wicked. So shall we then, being guilty of an error, even though it be wrong? We shall not then be blamed; there is no shame in what we try to do. These are all the laws that we invented by ourselves. From the beginning they proceeded independently. (…)
Temp. *t*=1.0, PPL =28.11	She walks in beauty. "Love," some old man says, "Belongs to two constant as those two stars." Beautiful diagonal line. So beautiful, the trees try to straighten it. "Wait a second," Peter says. "Is this exactly the last one?" For an example, let's suppose it's the last blue smoke. "Our Remains," Peter says. "How in the Hell's name is that supposed to be a song, though . . . " Won't this just be boring, you ask. Sure, says Peter, (…)
IQR-IP *p*=0.8, *k*=640, PPL =28.27 (*ours*)	She walks in beauty through all things good, as though a prince in the bloom of youth were ever born in any city. For this I would never forget the time I had spent with her, when we went through this temple. The perfume, the beautiful woman, the silence, the strange shadows, the pleasant voice, the flower of every description, were like those which now from her new cell perfume the fair shrine of Venus." \n And his memory fades into sleep, for at this very moment Venus rises from her silent chamber. The Roman fable has the goddess emerging from her palace in an instant from the black night of death. When the men are searching for her she rises from her throne, where the eyes of Death watch her silently, to welcome them. And from her presence a tumult is born, a struggle in darkness, a terrible din, of discordant cries. For this reason it was always sung, that if any were in a black room they should hear the shrill sound

level, traditional methods favor creating comparatively *plain and narrative* texts, while our algorithm favors creating *novel and surprising* texts. We relate these results to automatic results of diversity metrics by aggregating the filtered passages from all PPL levels per sampling algorithm to report their overall diversity metrics in Table 1. They show that under the on-equal-footing fluency paradigm, all methods are on par with each other regarding self-BLEU and 3-gram entropy. However, our method achieves a significantly lower Zipf coefficient, which confirms previous results. It reveals the nature of our method. Compared to traditional methods, our method dramatically flattens the word distribution of the generated texts (with a lower Zipf coefficient) by penalizing the high-likelihood words, which achieve similar fluency but exhibit higher diversity and novelty.

6 Conclusion

We propose the interquartile range inverse probability (IQR-IP) sampling algorithm for neural text generation. It rescales the high-likelihood part of the predicted distribution with inverse probability weighting, and conducts multi-filtering truncation on the low-likelihood part. Results show that our algorithm can significantly increase the diversity and novelty of the generated texts without corrupting the fluency.

References

1. Basu, S., Ramachandran, G.S., Keskar, N.S., Varshney, L.R.: MIROSTAT: a neural text decoding algorithm that directly controls perplexity. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=W1G1JZEIy5_
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(null), 1137–1155 (2003)
3. Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., Charlin, L.: Language GANs falling short. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=BJgza6VtPB
4. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 889–898. Association for Computational Linguistics, Melbourne (2018). https://doi.org/10.18653/v1/P18-1082, https://www.aclweb.org/anthology/P18-1082
5. He, T., Glass, J.: Negative training for neural dialogue response generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2044–2058. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.185, https://aclanthology.org/2020.acl-main.185
6. Hernán, M.A., Robins, J.M.: Causal Inference: What If. Chapman & Hall/CRC, Boca Raton (2020)
7. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=rygGQyrFvH
8. Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., Choi, Y.: Learning to write with cooperative discriminators. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1638–1649. Association for Computational Linguistics, Melbourne (2018). https://doi.org/10.18653/v1/P18-1152, https://www.aclweb.org/anthology/P18-1152
9. Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., Callison-Burch, C.: Comparison of diverse decoding methods from conditional language models. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3752–3762. Association for Computational Linguistics, Florence (2019). https://doi.org/10.18653/v1/P19-1365, https://aclanthology.org/P19-1365
10. Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. In: International Conference on Learning Representations (2017). https://openreview.net/forum?id=Byj72udxe

11. Nadeem, M., He, T., Cho, K., Glass, J.: A systematic characterization of sampling algorithms for open-ended language generation. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pp. 334–346. Association for Computational Linguistics, Suzhou (2020). https://aclanthology.org/2020.aacl-main.36
12. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
13. Shannon, C.E., Weaver, W.: A Mathematical Theory of Communication. University of Illinois Press, USA (1963)
14. Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., Weston, J.: Neural text generation with unlikelihood training. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=SJeYe0NtvH
15. Wolf, T., et al.: HuggingFace's transformers: state-of-the-art natural language processing. ArXiv abs/1910.03771 (2019)
16. Zhang, H., Duckworth, D., Ippolito, D., Neelakantan, A.: Trading off diversity and quality in natural language generation. In: Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval), pp. 25–33. Association for Computational Linguistics (2021). https://aclanthology.org/2021.humeval-1.3
17. Zhang, Y., et al.: Generating informative and diverse conversational responses via adversarial information maximization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018). https://proceedings.neurips.cc/paper/2018/file/23ce1851341ec1fa9e0c259de10bf87c-Paper.pdf
18. Zhu, Y., et al.: Texygen: a benchmarking platform for text generation models. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018, pp. 1097–1100. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3209978.3210080
19. Zipf, G.K.: Human Behaviour and the Principle of Least Effort. Addison-Wesley, Boston (1949)

Poster: Dialogue Systems

A Unified Generation Approach for Robust Dialogue State Tracking

Zijian Lin[1], Beizhang Guo[1], Tianyuan Shi[1], Yunhao Li[1], Xiaojun Quan[1(✉)],
and Liangzhi Li[2(✉)]

[1] School of Computer Science and Engineering, Sun Yat-sen University,
Guangzhou, China
quanxj3@mail.sysu.edu.cn

[2] Meetyou AI Lab (MAIL), Xiamen, China
liliangzhi@xiaoyouzi.com

Abstract. While dialogue state tracking by generation-based approaches allows for better scalability and generalization, they suffer from two major limitations. First, most generation-based models adopt a multi-task learning framework that may cause gradient conflicts and low training efficiency. Second, since the dialogue state of the previous turn is usually taken as an input for the current turn, there exists inconsistency between training and inference, which is identified as turn-level exposure bias. To address the first limitation, we propose the idea of state-transition sequence and transform multi-task learning into a single generation task. To alleviate turn-level exposure bias, we propose a slot-perturb strategy to reduce the over-reliance on the previous dialogue state. Experimental results show that our method achieves a new state of the art on the MultiWOZ 2.4 dataset and performs competitively on MultiWOZ 2.1. Besides, we demonstrate that the unified generation framework with slot-perturb improves the convergence speed and relieves error accumulation.

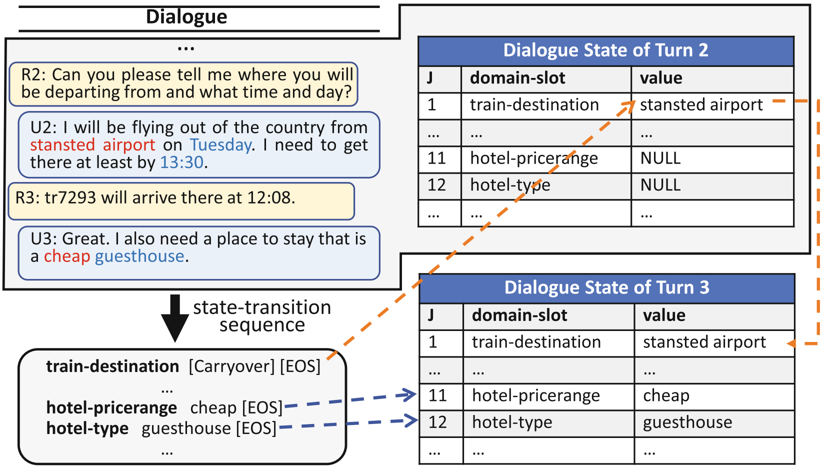
Keywords: Dialogue state tracking · task-oriented dialogue (TOD)

1 Introduction

Task-oriented dialogue (TOD) systems aim to accomplish various tasks such as attraction recommendation, hotel booking and travel planning through natural language conversations with human users [2]. One of the essential tasks in pipeline TOD systems is dialogue state tracking (DST), the goal of which is to predict user intentions at each turn of conversation and keep track of them during the whole dialogue. The dialogue state can be used to retrieve necessary entity information from external knowledge bases for the systems to generate appropriate responses. Typically, the dialogue state is defined as a set of <domain-slot, value> pairs (e.g., <*hotel-area, north*>) [1]. In Fig. 1, we present an example of dialogue state and the DST task.

Z. Lin and B. Guo are co-first authors and contribute equally to this work.

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 789–801, 2023.
https://doi.org/10.1007/978-3-031-44693-1_61



R2: Can you please tell me where you will be departing from and what time and day?

U2: I will be flying out of the country from stansted airport on Tuesday. I need to get there at least by 13:30.

R3: tr7293 will arrive there at 12:08.

U3: Great. I also need a place to stay that is a cheap guesthouse.

train-destination [Carryover] [EOS]
...
hotel-pricerange cheap [EOS]
hotel-type guesthouse [EOS]
...

Dialogue State of Turn 2

J	domain-slot	value
1	train-destination	stansted airport
...	...	...
11	hotel-pricerange	NULL
12	hotel-type	NULL
...	...	...

Dialogue State of Turn 3

J	domain-slot	value
1	train-destination	stansted airport
...	...	...
11	hotel-pricerange	cheap
12	hotel-type	guesthouse
...	...	...

Fig. 1. Dialogue state tracking by our approach. Words in red/blue represent new values that appear at the current turn. (Color figure online)

Most traditional DST methods follow the classification paradigm [10,22], which tries to select a candidate value in the pre-defined ontology to complete the task. However, these methods have inherent limitations. First, it is impossible to predict the values that are not included in the ontology at inference time. Second, to predict the current dialogue state, these methods have to go over all slot-value candidates at every turn, which is inefficient [8]. More importantly, the ontology often changes dynamically in real-life scenarios, meaning that a fixed ontology is not flexible and adaptive [7].

To tackle these issues, generation-based approaches treat DST as a sequence generation task, where the value of each slot is modeled as the target sequence to be generated [18]. When generating a value sequence, these models output the distribution of each token over vocabulary without pre-defining the ontology. Thus, unseen values can be handled at inference time, allowing for greater flexibility and scalability.

Most existing generation-based DST models adopt multi-task learning, i.e., jointly training a slot classifier and a value generator [8,18,21]. However, such an approach may lead to sub-optimal network weights when the sub-task gradients conflict or are dominated by one of the tasks [16]. To address this issue, we propose the idea of *state-transition sequence* to model both slot classification and value generation in a unified generation framework. As shown in Fig. 1, for each slot the model only needs to generate a state-transition sequence without performing slot classification, thus transforming multi-task learning into single-task learning and eliminating potential gradient conflicts. Moreover, the generator is expected to be enhanced in this unified framework.

Moreover, the dialogue state of the previous turn is usually used to represent the dialogue history when constructing the input sequence, which not only abandons non-immediate information [23] but also provides direct and compact representation for the models [8]. While the training phase uses the ground truth slot values, the inference phase uses the generated slot values, leading to a discrepancy between training and inference which we call turn-level exposure bias. As the number of dialogue turns increases, errors inevitably accumulate. To alleviate this problem, we propose the *slot-perturb* strategy, which aims to train the model by stimulating the condition it will face at inference time. Firstly, we define a probability p to decide whether to perturb the previous dialogue state, which increases linearly as the dialogue turns grow. Secondly, we choose a set of slots in the previous dialogue state based on two rules. Thirdly, we apply two types of perturbation to replace the values of the chosen slots.

We evaluate the proposed model on two large-scale datasets, MultiWOZ 2.4 and MultiWOZ 2.1. Experimental results show that this model achieves state-of-the-art performance on MultiWOZ 2.4 and competitive performance on MultiWOZ 2.1. Moreover, we demonstrate that our unified generation approach eliminates gradient conflicts and boosts the convergence speed. Finally, we conduct an in-depth analysis of the slot-perturb strategy and illustrate that it mitigates turn-level exposure bias.

2 Related Work

2.1 Dialog State Tracking

Traditional approaches to DST are mostly classification-based [10,17,22], which predict slot values in a pre-defined ontology. Nevertheless, the requirement of a fixed ontology could be unavailable in some scenarios. To overcome this problem, various generative approaches are studied. Wu et al. [18] employ a bidirectional GRU to encode the dialogue history and a copy-based GRU to decode the value of each slot. Kim et al. [8] take the dialogue state of the previous turn and dialogue history as input to a BERT encoder and employ a copy-based GRU as the decoder. These approaches generally decompose DST into two sub-tasks, which may result in sub-optimal network weights as gradients conflict [16]. Zeng et al. [21] try to solve this problem by jointly optimizing state operation prediction and value generation, yet they still model DST as two sub-tasks.

The above approaches can be improved by using large-scale dialogue data to pre-train the models [11,12,15]. Mehri et al. [11] fine-tune a BERT model on a large open-domain dialogue corpus to better model dialogues. Peng et al. [12] model DST using a Transformer-based auto-regressive language model and pre-train it on heterogeneous dialogue data. Su et al. [15] propose a multi-task pre-training strategy on heterogeneous dialogue corpora. Unlike them, our model does not rely on extra data for pre-training but only utilizes the previous dialogue state and the dialogue history.

2.2 Exposure Bias

The dialogue state of the previous turn can be used as a direct and compact representation of the dialogue history [8]. However, the input dialogue state could be incorrect at inference time, which brings a gap between training and inference. We follow the notion of exposure bias in text generation and identify this problem as turn-level exposure bias in DST. Exposure bias [13] has been widely studied in language generation. Schmidt et al. [14] attribute it to the generalization gap caused by distribution and domain shift.

3 Methodology

Figure 2 shows the architecture of our model. We first introduce the settings and then introduce our model and the slot-perturb strategy to alleviate turn-level exposure bias.

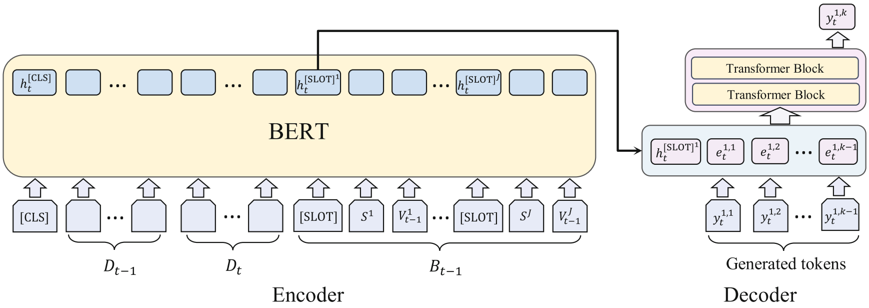


Fig. 2. The architecture of our unified generation model for DST. This figure shows how to generate a state-transition sequence for the first slot. A slot-perturb strategy is applied to B_{t-1} to alleviate the turn-level exposure bias issue.

3.1 Problem Setting

Dialogue State Tracking. Let $D_t = \{R_t, U_t\}$ represents the conversation between system and user at turn t, where R_t denotes the system response and U_t denotes the user utterance of this turn. We use $B_t = \{(S^j, V_t^j)|1 \leq j \leq J\}$ to denote the dialogue state for turn t, where S^j and V_t^j are the j-th domain-slot and value, respectively, and J is the total number of domain-slots. For simplicity, we occasionally use *slot* to represent *domain-slot* throughout this paper. Following [8], we only use $\{(D_{t-1}, B_{t-1}), D_t\}$ to represent the dialogue context. Two special values [NULL] and [DONTCARE] are defined, which means the slot has null value and the user does not care what the corresponding value is, respectively.

State-Transition Sequence. We model DST as a sequence generation task, in which a state-transition sequence Y_t^j is defined to describe the change of the value of slot S^j between two adjacent turns $t-1$ and t. Specifically, given the values of V_t^j and V_{t-1}^j for slot S^j from two adjacent turns, we define the state-transition sequence Y_t^j as:

$$Y_t^j = \begin{cases} \text{[CarryOver]} \oplus \text{[EOS]}, & \text{if } CO \\ \text{[Delete]} \oplus \text{[EOS]}, & \text{if } DEL \\ \text{[Dontcare]} \oplus \text{[EOS]}, & \text{if } DC \\ V_t^j \oplus \text{[EOS]}, & \text{if } UPD \end{cases} \tag{1}$$

where [EOS] is a special token used to mark the end of the state-transition sequence, and $\oplus$ is used to denote the concatenation of two sequences. CO, DEL, DC, and UPD are four operations to be defined in the following paragraphs.

Operations. We define the above four operations, namely, update (UPD), delete (DEL), don't care (DC) and carryover (CO), for slot values between two adjacent turns:

$$UPD \rightarrow V_t^j \neq V_{t-1}^j \tag{2}$$

$$DEL \rightarrow V_t^j = \text{[NULL]} \ \&\ V_{t-1}^j \neq \text{[NULL]} \tag{3}$$

$$DC \rightarrow V_t^j = \text{[DONTCARE]} \tag{4}$$

$$CO \rightarrow V_t^j = V_{t-1}^j \tag{5}$$

At inference time, a state-transition sequence $Y_t^j = [y_t^{j,1}, y_t^{j,2}, ..., y_t^{j,N}]$ is firstly generated for slot S^j at turn t, where $y_t^{j,1}$ represents the first token of Y_t^j, and $y_t^{j,N}$ is the [EOS] token. Then, we recover the value V_t^j for this slot according to Y_t^j and V_{t-1}^j.

$$V_t^j = \begin{cases} V_{t-1}^j, & \text{if } y_t^{j,1} = \text{[CarryOver]} \\ \text{[NULL]}, & \text{if } y_t^{j,1} = \text{[Delete]} \\ \text{[DONTCARE]}, & \text{if } y_t^{j,1} = \text{[Dontcare]} \\ [y_t^{j,1}, ..., y_t^{j,N-1}], & \text{otherwise} \end{cases} \tag{6}$$

The idea of state-transition sequence is inspired by the observation that the dialogue state only changes slightly between two adjacent turns. Therefore, comparing to the approach of generating a sequence of real slot values, the above approach simplifies the task as most of the tokens can be selected from [CarryOver], [Delete] and [Dontcare]. Then, the new dialogue state can be derived from state-transition sequences easily.

3.2 Context Encoder

We first obtain a sequence $\hat{D}_t = R_t \oplus ; \oplus U_t \oplus \text{[SEP]}$ to represent conversation D_t, where ; is a special token marking the demarcation of two sequences and [SEP] marks the end of this sequence. Similarly, we represent the dialogue state B_t of turn t by a sequence $\hat{B}_t = B_t^1 \oplus B_t^2 \oplus ... \oplus B_t^J$, where $B_t^j = \text{[SLOT]} \oplus S^j \oplus - \oplus V_t^j$ denotes the j-th slot-value pair of this turn, in which [SLOT] is used to mark the start of a new <domain-slot, value> pair and $-$ is a special token to connect S^j and V_t^j. The encoder of our model is a pre-trained BERT [4], whose input is the concatenation of $\hat{D}_{t-1}$, $\hat{D}_t$ and $\hat{B}_{t-1}$:

$$X_t = \text{[CLS]} \oplus \hat{D}_{t-1} \oplus \hat{D}_t \oplus \hat{B}_{t-1}, \tag{7}$$

where [CLS] is a special token used to identify the beginning of the input sequence.

The output of this encoder is represented as $H_t \in \mathbb{R}^{|X_t| \times d}$, where $|X_t|$ denotes the length of X_t and d is the dimension of hidden states. Additionally, we use $h_t^{[\mathrm{SLOT}]^j} \in \mathbb{R}^d$ in H_t to represent the hidden state of the j-th [SLOT] token at turn t, which incorporates necessary contextual information.

3.3 The Generator

We employ a decoder of two Transformer layers as the generator to autoregressively generate a state-transition sequence for each slot. For the j-th slot at turn t, the generator is initialized with $e_t^{j,0} = h_t^{[\mathrm{SLOT}]^j}$ and recurrently outputs the hidden state $g_t^{j,k}$ at the k-th decoding step by feeding the embeddings $\{e_t^{j,i}\}_{i=0}^{k-1}$ until the [EOS] token is decoded. Here, $e_t^{j,i}$ is the combination of word and position embeddings of token $y_t^{j,i}$:

$$e_t^{j,i} = \mathrm{Emb}(y_t^{j,i}) + \mathrm{Pos}(y_t^{j,i}), \tag{8}$$

$$g_t^{j,k} = \mathrm{Decoder}(H_t, \{e_t^{j,i}\}_{i=0}^{k-1}), \tag{9}$$

where $\mathrm{Emb}(\cdot)$ and $\mathrm{Pos}(\cdot)$ are the token and position embedding functions, respectively, and $\mathrm{Decoder}(\cdot)$ denotes the decoder. Note that the word and position embedding matrices of the decoder are shared with the encoder.

Then, $g_t^{j,k}$ is converted into a probability distribution over vocabulary to predict the k-th token:

$$P_t^{j,k} = \mathrm{softmax}(W g_t^{j,k}), \tag{10}$$

where $W \in \mathbb{R}^{L \times d}$ is a linear transformation matrix that is shared with the word embedding matrix of the encoder and L is the size of the vocabulary.

3.4 Turn-Level Exposure Bias

We seek to alleviate the turn-level exposure bias problem by presenting a slot-perturb strategy. As shown in Fig. 3, the general idea is to train the model in simulated situations that might arise at inference time, so that we can reduce the gap between training and inference. Specifically, we divide this strategy into three steps.

Turn-Select. We set a probability p to decide whether to perturb the last dialogue state B_{t-1} in the current input. We use the perturbed dialogue state B_{t-1}^* to replace B_{t-1}. Moreover, the problem of incorrect predictions in DST typically becomes severe as the number of dialogue turns increases. To take this fact into account, we allow p to increase linearly with the number of dialogue turns:

$$p = \begin{cases} \alpha T, & \alpha T < 1 \\ 1, & \text{otherwise}, \end{cases} \tag{11}$$

where α is a factor to determine the increase rate of perturbation and T is the number of dialogue turns.

Slot-Select. We apply two principles to select potential slots. First, the slots whose values could be inferred from the dialogue history are selected since the model either samples from the previous dialogue states or the dialogue history to extract the current dialogue states. If the previous dialogue states are modified by slot perturbation, the correct value must be able to be inferred from dialogue history. Second, the slots whose values will not be updated at the current turn are selected, expecting the model to focus on the information in the dialogue history to correct the wrong values in previous dialogue states.

We denote the overall slot-select operation as:

$$S^*_{t-1} = \text{SLOT\_SEL}(\{S^j\}^J_{j=1}), \tag{12}$$

where S^*_{t-1} is the final set of selected slots. Then, we perturb the values of slots in S^*_{t-1} as follows.

Value-Replace. For each slot in S^*_{t-1} at the t-th turn, we apply two types of perturbations to change its value from V^j_{t-1} to V^{*j}_{t-1}. First, we can replace V^j_{t-1} with a randomly sampled value from the candidate values of this slot to force the sampled value to be different from the current one, which simulates incorrect value generation. Second, we can replace V^j_{t-1} with a special token [NULL] to simulate undergeneration. For each slot in S^*_{t-1}, we apply either of the two types of perturbations with the same probability.

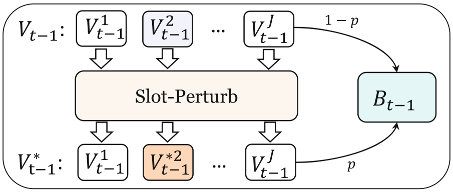


Fig. 3. The framework of our slot-perturb strategy.

3.5 Objective

For each turn of conversation, we employ standard cross-entropy to calculate the loss of the predicted dialogue state:

$$\mathcal{L} = -\frac{1}{J \times K} \sum_{j=1}^{J} \sum_{k=1}^{K} (y^{j,k}_{label,t})^T \log P^{j,k}_t$$

where J is the total number of slots and K is the length of a generated state-transition sequence. Moreover, $y^{j,k}_{label,t}$ is the one-hot vector of the k-th ground truth token for the j-th slot.

4 Experiments

In this section, we introduce the evaluation details of our generation model, namely TransDST.

Datasets. We evaluate TransDST on MultiWOZ 2.1 [5] and MultiWOZ 2.4 [19]. Since MultiWOZ 2.1 suffers from incorrect and inconsistent annotations in the validation and test sets [22], MultiWOZ 2.4 was released to fix these errors. We follow the preprocessing strategy of [18][1]. Following previous work [18], we only use five domains (*restaurant*, *hotel*, *attraction*, *taxi*, and *train*) in our experiments because the domains of *hospital* and *police* are not included in the validation and test sets.

Training. We use BERT-base-uncased and BERT-large-uncased as encoder and set the peak learning rate between [1e-5, 4e-5] for the encoder, and [8e-6, 3e-5] for the decoder. We set the batch size to 16. Both the encoder and decoder have the same hidden size of 768, with Adam as the optimizer. The warmup proportion and dropout rate are both set to 0.1. Our implementation is based on PyTorch and the HuggingFace Transformers toolkit. For the slot-perturb strategy, we set the factor α to 0.1.

Baselines. For comparison, we select a variety of baseline systems from previous literature. (1) Classification-based: TripPy [7], SAVN [17], SUMBT [10], DS-Picklist [22], and STAR [20]. (2) Generation-based: TRADE [18], PIN [3], SOM-

Table 1. Overall results on MultiWOZ 2.1 and MultiWOZ 2.4.

Model	JGA (%)	
	MWOZ 2.1	MWOZ 2.4
Classification-based Approaches		
TripPy [7]	55.29	59.60
SAVN [17]	54.86	60.55
SUMBT [10]	49.01	61.86
DS-Picklist [22]	53.30	-
STAR [20]	**56.36**	73.62
Generation-based Approaches		
TRADE [18]	46.00	55.10
PIN [3]	48.40	58.92
SOM-DST [8]	53.68	66.78
Seq2Seq-DU [6]	56.10	-
SDP-DST [9]	**56.66**	-
TransDST (BERT-base)	55.24	72.05
TransDST (BERT-large)	55.90	**73.81**

[1] https://github.com/jasonwu0731/trade-dst.

DST [8], Seq2Seq-DU [6], and SDP-DST [9]. As our model is only fine-tuned on the target dataset, we do not compare it with models trained on augmented or external corpora to make the comparison fair.

4.1 Overall Results

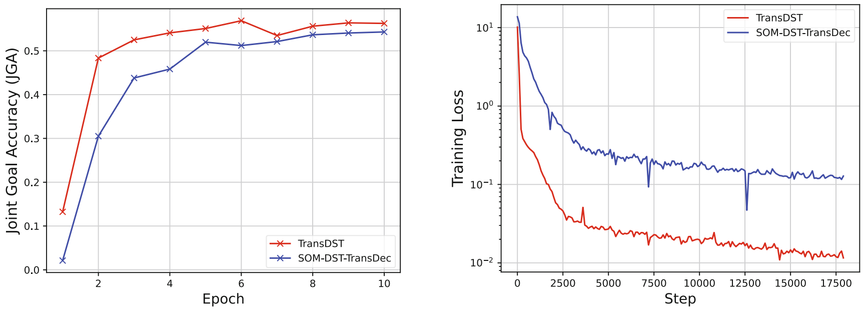
Follow prior works, the evaluation metric we employ is joint goal accuracy (JGA), which is defined as the percentage of turns in which all the slots are predicted correctly. The results are shown in Table 1.

Results on MultiWOZ 2.4. TransDST achieves state-of-the-art performance among all the generation-based models on the MultiWOZ 2.4 test set. TransDST with BERT-large-uncased even outperforms existing classification-based models, which have been studied relatively longer and have strong performance. Moreover, compared to the previous state-of-the-art generation-based model, SOM-DST, our model with BERT-base-uncased achieves a significant improvement of 5.27 points, confirming the effectiveness of our unified generation framework.

Results on MultiWOZ 2.1. On the MultiWOZ 2.1 test set, our model achieves competitive performance compared to the baselines. It is reasonable that its performance drops on MultiWOZ 2.1 compared to the result on MultiWOZ 2.4, as the noisy data in MultiWOZ 2.1 influences the performance. Although Seq2Seq-DU and SDP-DST produce better results, Seq2Seq-DU utilizes an extra schema in the input, and if slot values are categorical, it can directly point to the values in the schema. SDP-DST also relies on additional prompt information to improve the performance. Besides, the performance of STAR surpasses TransDST on MultiWOZ 2.1 but underperforms TransDST on MultiWOZ 2.4, showing the potential of TransDST on clean data.

4.2 Ablation Study

We conduct an ablation experiment to investigate how the modules in our proposed approach affect DST performance. In particular, we focus on investigating



(a) (b)

Fig. 4. (a) Performance of TransDST and SOM-DST-TransDec on the validation set of MultiWOZ 2.1. (b) Loss of TransDST and SOM-DST-TransDec on the training set of MultiWOZ 2.1. Note that the logarithmic scale is applied for the y-axis.

two situations in this experiment. While one is whether to adopt state-transition sequence to unify slot classification and value generation, the other is whether to apply the slot-perturb strategy during training.

Table 2. Results of ablation study of our TransDST, where *S-T sequence* denotes state-transition sequence and *S-P Strategy* denotes the slot-perturb strategy.

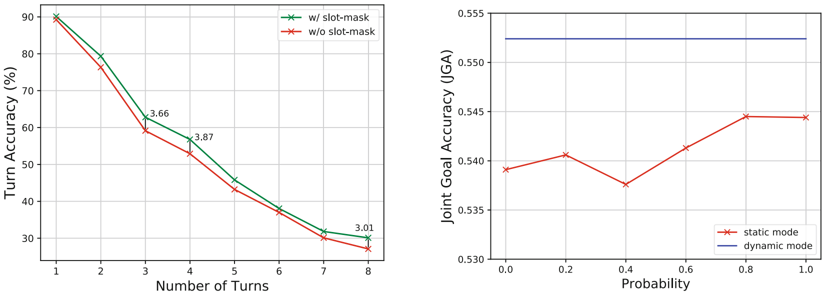
S-T sequence	S-P strategy	JGA (%)
w/o	w/o	52.27
w/o	w/	53.84
w/	w/o	53.91
w/	w/	55.24

We observe from Table 2 that replacing single-task learning with multi-task learning hurts the joint goal accuracy, showing the effectiveness of state-transition sequence. Moreover, in the following subsection we will demonstrate that the training efficiency can be greatly boosted by converting multi-task training to single-task training using the idea of state-transition sequence. Besides, we observe that the slot-perturb strategy can also increase the joint goal accuracy in a multi-task learning setting, showing the generality of this strategy.

4.3 Analysis of State-Transition Sequence

State-transition sequence allows our model to integrate slot classification and value generation in a unified generation task, eliminating potential gradient conflicts. We further show the advantages of state-transition sequence on MultiWOZ 2.1.

JGA on Validation Set. We compare the joint goal accuracy of TransDST and SOM-DST on the validation set of MultiWOZ 2.1. For a fair comparison,



(a) (b)

Fig. 5. (a) Results of turn accuracy of TransDST trained with and without the slot-perturb strategy. The differences between them at certain turns are provided. (b) Joint goal accuracy (JGA) of TransDST trained with slot-perturb in dynamic and static.

we replace the GRU decoder of SOM-DST with TransDST's decoder and denote the variant as SOM-DST-TransDec, which has the same scale of parameters as TransDST. Figure 4a shows the results of TransDST and SOM-DST-TransDec. Overall, TransDST outperforms SOM-DST-TransDec in every epoch, proving the consistent effectiveness of state-transition sequence. Moreover, in the first several epochs TransDST performs much better than SOM-DST-TransDec. We speculate this is because the gradients of the two sub-tasks conflict severely at the beginning, leading to poor performance and training efficiency. On the contrary, state-transition sequence transforms multi-task learning into single-task learning, avoiding gradient conflicts and leading to better results.

Loss on Training Set. We further show the mean loss of TransDST and SOM-DST-TransDec on the training set of MultiWOZ 2.1. As plotted in Fig. 4b, at the early stage of training, the loss of TransDST drops rapidly while the loss of SOM-DST-TransDec drops more slowly. Moreover, after certain steps of training, the loss of SOM-DST-TransDec is almost an order of magnitude higher than the loss of TransDST. As a result, SOM-DST-TransDec requires 30 epochs to obtain acceptable performance, while TransDST only needs 10 epochs for training to achieve desirable results. This experiment demonstrates that TransDST significantly outperforms SOM-DST-TransDec in convergence speed, showing the advantage of combining slot classification and value generation into a single state-transition sequence generation task.

4.4 Analysis of the Slot-Perturb Strategy

Exposure Bias. To confirm that the slot-perturb strategy is able to alleviate turn-level exposure bias, we select the dialogues including at least β turns of conversations from the test set of MultiWOZ 2.1 as a new test set. For the dialogues which contain more than β turns of conversations, we clip it to β turns. Then, we define turn accuracy TA_t for the t-th turn of the new test set as:

$$\mathrm{TA}_t = \frac{\#\mathrm{correct}_t}{\#\mathrm{total}_t}, \tag{13}$$

where $\#\mathrm{correct}_t$ denotes the number of conversations at the t-th turn in which all the slots are predicted correctly, and $\#\mathrm{total}_t$ is the total number of conversations at the t-th turn in the test set.

As shown in Fig. 5a, where β is equal to 8, TransDST trained with the slot-perturb strategy achieves consistently superior performance. With the number of turns increases, the difference reaches its peak, which is nearly 4%. This result demonstrates the effectiveness of our approach to a certain degree. It also means that our model can generate the sequences more accurately at each turn regardless of the correctness of the previous dialogue state, which indicates this approach can alleviate turn-level exposure bias and relieve error accumulation. It is noticeable that the difference becomes large in later turns of dialogues, which demonstrates the slot-perturb strategy can improve the performance of the model in a long dialogue.

Dynamic vs. Static Turn-Select. As introduced in Sect. 3.4 and Eq. (11), we opt for a dynamic mode to set the turn-select probability, rather than a static one. The static mode means using the same probability throughout all dialogue turns to determine if the input dialogue state needs to be perturbed. To demonstrate the effectiveness of the dynamic mode, we experiment on the test set of MultiWOZ 2.1. We run our model in a static mode with the probability from 0 to 1 stepped by 0.2. As shown in Fig. 5b, the slot-perturb strategy in dynamic mode consistently produces superior performance to the static one. Moreover, in a static mode the model can still achieve better performance than the one trained without the slot-perturb strategy.

5 Conclusion

We present a unified generation model, TransDST, for DST. TransDST transforms multi-task learning into a single generation task and has the merits of eliminating gradient conflicts and improving training efficiency. Moreover, we identify the turn-level exposure bias problem in DST and propose the slot-perturb strategy to relieve it. Experimental results show that our method achieves a new state of the art on the MultiWOZ 2.4 dataset and performs competitively on MultiWOZ 2.1. Besides, we demonstrate that the unified generation framework with slot-perturb improves the convergence speed and relieves error accumulation.

References

1. Budzianowski, P., et al.: MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In: Proceedings of EMNLP (2018)
2. Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: recent advances and new frontiers. SIGKDD Explor. Newsl. **19**(2), 25–35 (2017)
3. Chen, J., Zhang, R., Mao, Y., Xu, J.: Parallel interactive networks for multi-domain dialogue state generation. In: Proceedings of EMNLP (2020)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, Volume 1 (Long and Short Papers) (2019)
5. Eric, M., et al.: MultiWOZ 2.1: a consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In: Proceedings of LREC, pp. 422–428 (2020)
6. Feng, Y., Wang, Y., Li, H.: A sequence-to-sequence approach to dialogue state tracking. In: Proceedings of ACL/IJCNLP (2021)
7. Heck, M., et al.: TripPy: a triple copy strategy for value independent neural dialog state tracking. In: Proceedings of SIGDIAL (2020)
8. Kim, S., Yang, S., Kim, G., Lee, S.W.: Efficient dialogue state tracking by selectively overwriting memory. In: Proceedings of ACL (2020)
9. Lee, C.H., Cheng, H., Ostendorf, M.: Dialogue state tracking with a language model using schema-driven prompting. In: Proceedings of EMNLP, pp. 4937–4949 (2021)
10. Lee, H., Lee, J., Kim, T.Y.: SUMBT: slot-utterance matching for universal and scalable belief tracking. In: Proceedings of ACL, pp. 5478–5483 (2019)

11. Mehri, S., Eric, M., Hakkani-Tur, D.: DialoGLUE: a natural language understanding benchmark for task-oriented dialogue. arXiv e-prints arXiv:2009.13570 (2020)
12. Peng, B., Li, C., Li, J., Shayandeh, S., Liden, L., Gao, J.: SOLOIST: building task bots at scale with transfer learning and machine teaching. Trans. Assoc. Comput. Linguis. **9**, 807–824 (2021)
13. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. arXiv e-prints arXiv:1511.06732 (2015)
14. Schmidt, F.: Generalization in generation: a closer look at exposure bias. In: Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 157–167 (2019)
15. Su, Y., et al.: Multi-task pre-training for plug-and-play task-oriented dialogue system. In: Proceedings of ACL, pp. 4661–4676 (2022)
16. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3614–3633 (2022)
17. Wang, Y., Guo, Y., Zhu, S.: Slot attention with value normalization for multi-domain dialogue state tracking. In: Proceedings of EMNLP (2020)
18. Wu, C.S., Madotto, A., Hosseini-Asl, E., Xiong, C., Socher, R., Fung, P.: Transferable multi-domain state generator for task-oriented dialogue systems. In: Proceedings of ACL, pp. 808–819 (2019)
19. Ye, F., Manotumruksa, J., Yilmaz, E.: MultiWOZ 2.4: a multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. CoRR abs/2104.00773 (2021)
20. Ye, F., Manotumruksa, J., Zhang, Q., Li, S., Yilmaz, E.: Slot self-attentive dialogue state tracking. In: Proceedings of WWW, pp. 1598–1608 (2021)
21. Zeng, Y., Nie, J.Y.: Multi-domain dialogue state tracking-a purely transformer-based generative approach. CoRR abs/2010.14061 (2020)
22. Zhang, J., et al.: Find or classify? Dual strategy for slot-value predictions on multi-domain dialog state tracking. In: Proceedings of COLING, pp. 154–167 (2020)
23. Zhao, J., Mahdieh, M., Zhang, Y., Cao, Y., Wu, Y.: Effective sequence-to-sequence dialogue state tracking. In: Proceedings of EMNLP, pp. 7486–7493 (2021)

An Explicit-Memory Few-Shot Joint Learning Model

Fanfan Du[1], Meiling Liu[1](✉), Tiejun Zhao[2], and Shafqat Ail[2]

[1] Northeast Forestery University, Harbin 150000, China
{2453999566,mlliu}@nefu.edu.cn
[2] Harbin Institute of Technology, Harbin 150000, China
{tjzhao,sali406}@hit.edu.cn

Abstract. There are two difficulties in existing spoken language understanding models. The first problem is that it is difficult to extract the implicit relationship information between the intention and the slot in the utterance for the inference process, and the inference effect is not ideal; the second problem is that the training data is scarce, and the existing models cannot learn from the small amount of training data. Get more useful information. To address these two challenges, this paper proposes an Explicit-Memory Few-shot join learning model. In order to solve the first problem, a multi-layer model structure from coarse to fine is adopted to train the hidden semantic relationship and hidden state information between intentions and slots in the utterance; in order to solve the second problem, using the Siamese BERT metric learning method to jointly train the model. We use the Snips and ATIS datasets to train the model, and the test results show better results. In the case of a small amount of data, the model can also obtain stronger inference ability.

Keywords: Intent detection · Slot filling · Explicit memory · Few-shot

1 Introduction

Spoken language understanding plays a crucial role in intelligent voice dialog in task-based human-machine intelligent voice dialog applications such as voice assistants Alexa, Siri, and Cortana. These applications automatically recognize the intent of the user by completing the two tasks of intent detection and slot filling in the spoken language understanding module, Fig. 1 show an example of intent detection and slot filling "Put 'sungmin' into my summer playlist." Slot filling assigns a slot label to each word, and intent detection matches the intent label for the entire sentence and shows the associations between them.

Using large-scale data in deep neural networks, such as recurrent neural networks (RNNs) (especially gated recurrent unit (GRU)) and long short-term memory (LSTM) models, have been extensively used for studying intent detection and slot filling [1–4]. Introducing an attention mechanism [5] helps recurrent neural networks to deal with a long-term dependent state. a joint modeling method based on attention recurrent neural networks has been proposed [3].

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 802–813, 2023.
https://doi.org/10.1007/978-3-031-44693-1_62

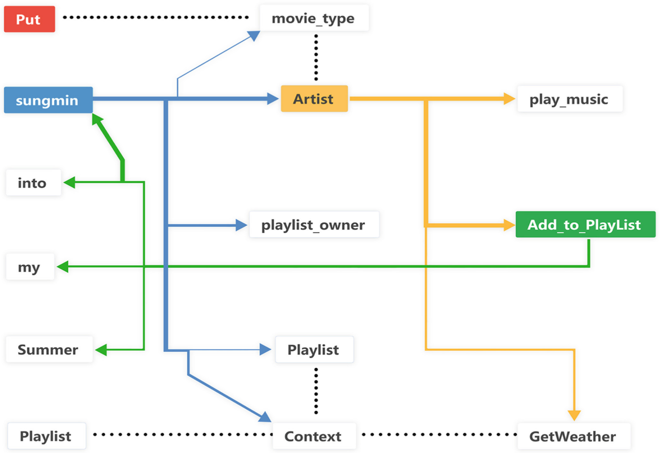


Fig. 1. Interaction statistics of sentences, slot labels, and intent labels. Lines with different colors indicate the association between different components, and the width indicates the degree of association.

Memory networks are the first proposed memory nteworks, which use memory components to store scene information to realize the function of long-term memory. For many neural network models, RNN, LSTM and its variant GRU use a certain memory mechanism.

The size of training samples may be small due to difficulties in data collection and high cost of sample labeling. Therefore, the problem of small samples has become one of the important research directions in the field of machine learning.

However, existing methods suffer from two major drawbacks. First, Due to the difficulty of data collection and the high cost of sample annotation, the size of training samples is small, few-shot learning models can learn multiple tasks from few-show samples. Second, When there is implicit state in the discourse, implicit relational reasoning is needed to show the intention, existing methods cannot explicitly track the hidden states of intents and slots in user-asked questions to efficiently reason about their implicit relationships for better decision-making.

The contributions of this paper can be listed as follows:

1. An explicit-memory few-shot joint learning model based on Siamese BERT was developed that can obtain the association between tasks in multiple samples, perform multi-task learning, and obtain more information from a small number of samples. The proposed model helps overcome the problem encountered in traditional methods wherein many intent and slot features are required for obtaining the similarity features of different sentences.
2. An explicit memory structure was developed that can track the hidden semantic state between the intention and the slot in the utterance in an explicit manner, update the sentence state, and then extract the relevant segment for implicit relationship.

3. The proposed model achieves state-of-the-art results on two datasets (SNIPS and ATIS), demonstrating the effectiveness and generalization ability of the model for spoken language understanding tasks.

2 Related Works

At present, Few-Shot Learning (FSL) has achieved great success in many fields of deep learning, such as image classification, text classification, etc. It has now been extended to intent detection and slot filling. For example, intent and slot representations are extracted through bidirectional interaction, and the prototype network is extended to achieve explicit joint learning [6]. Data augmentation is performed with a conditional generator that is trained directly with the meta-learning objective [7]. Recently, pre-trained language models (PLMs) have emerged as a simple yet promising solution to a wide spectrum of natural language processing (NLP) tasks, triggering the surge of PLM-based solutions for few-shot intent detection, which typically fine-tune PLMs on conversation data [8].Isotropization techniques, which yield significant performance improvement in many tasks [9], have led to the proposal of two simple and effective isotropization regularizers [10].

For the intent detection model, [11] used convolutional neural networks to extract text vectors. This method of feature recognition can be used for deeper speech extraction. [12] proposed to apply a recurrent neural network to intent detection, which can use context information to improve its accuracy. [13] proposed to use LSTM and RNN for intent detection. This verifies that the deep learning model is effective in intent detection, leading to a significant improvement.

Considering the strong semantic correlation between slots and intents, some studies propose performing unidirectional augmentation by correlating both intents and slots [4,14]. Intuitively, slot filling also has instructive value for the intent detection task. At the same time, [15] proposed a layered capsule network to perform cross-interaction among words, slots, and intents in a pipelined manner.

The SBert proposed by [16] in 2019 follows the structure of the Siamese network by SBERT. The two Sentence Encoders use the same BERT and then add a pooling (pooling) operation to achieve the output Sentence vectors of the same size. In the output layer, the similarity between the two vectors can be compared by methods such as cosine similarity, which is capable of solving the problem of small samples, so we use SBert to encode.

3 Approach

As shown in Fig. 2, the model proposed in this study consists of the following four main modules.

1. Encoder: It uses SBERT [16] to concatenate the intent and slot of the input utterance question into a contextualized representation.

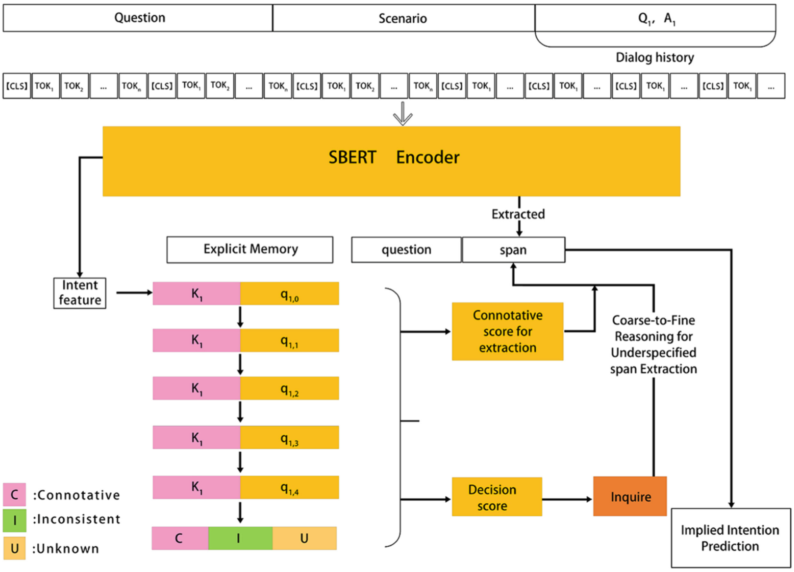

Question
Scenario
Q1, A1
Dialog history
[CLS] TOK1 TOK2 ... TOKn [CLS] TOK1 TOK2 ... TOKn [CLS] TOK1 TOK2 ... TOKn [CLS] TOK1 ... [CLS] TOK1 ... [CLS] TOK1 ... [CLS] TOK1 ...
SBERT Encoder
Extracted
Explicit Memory
question
span
Intent feature
K1 q1,0
K1 q1,1
K1 q1,2
K1 q1,3
K1 q1,4
C I U
Connotative score for extraction
Coarse-to-Fine Reasoning for Underspecified span Extraction
Decision score
Inquire
Implied Intention Prediction
C :Connotative
I :Inconsistent
U :Unknown

Fig. 2. The proposed model includes the following processes: SBERT encoding, explicit memory of sentence semantic implicit state, reasoning decision, and implicit intent prediction to reveal all utterance intents.

2. Explicit Memory: It reads the user question, predicts the hidden state (the state that is not explicitly mentioned) of the semantic relationship between the intention and the slot that is not specifically expressed in each sentence, and updates the hidden state of the sentence.
3. Inference Decision: It performs implicit relationship reasoning between slots and intents according to the updated state of the sentence.
4. Implicit Intent Prediction: If the decision is "Inquire," then the Implicit Intent Prediction module utilizes the implicit state of the semantics in the sentence and extracts the segment with the largest amount of information in the dialog in a coarse-to-fine manner to identify all the intentions of the end-user.

4 Encording

Let f_K represent the utterance. First, the utterance f_K is segmented, and the [CLS] token is inserted at the beginning of the sentence according to the sentence boundary or condition.

Then, SBERT [16], which is a pretrained encoder [17] that can encode the sequence of tokens into a sequence of vectors with the same length, is used. The token representation of each sentence [CLS] is regarded as the feature representation of the input sentence; thus, the token-level and sentence-level representations of each sentence are obtained. The sentence level of the dialog can be expressed as $k_1, \ldots, k_M$, and the token level can be expressed as

$[(f_{1,1},\ldots,f_{1,n_1}),\ldots,(f_{M,1},\ldots,f_{M,n_M})]$, where n_i is the number of tokens of sentence i.

5 Explicit Memory

The information provided by the user is the utterance s_K. In order to understand the relevant information of the question more quickly, our model tracks the hidden state between the intention and slot in the information provided by the discourse in the way of display. For example, if a similar question has been asked before, but the user did not express it completely, the model will analyze it in time and find the answer to be answered more quickly through memory. [15] proposed a reroute pattern to infer intentions. Inspired by the reroute pattern, we proposed the Display memory Module (EM), a gated memory neural network that understands user questions sequentially and updates the information state hidden by tracking conversations. We proposed the Explicit Memory module, which is a gated memory neural network that updates the information state hidden by the tracked dialog by sequentially understanding the dialogue information provided by the user.

As shown in Fig. 2, the Explicit Memory module saves the updated semantic state information in an explicit manner. The value of each state q_i matches the value of the corresponding sentence for initialization. Then, the model reads the sentence information $Q_K, Q_Q, Q_1, \ldots, Q_P$ provided by the utterance. When the time length is t, the sentence information $s_t \in \{Q_K, Q_Q, Q_1, \ldots, Q_P\}$ provided by the user is used to update the intent of the i-th sentence and the semantic value state $q_{i,t}$ implied by the slot.

$$\tilde{q}_{i,t} = V_k + V_q q_{i,t} + V_Q Q_t \tag{1}$$

$$e_i = \sigma\left(Q_t^T + Q_t^T q_i\right) \in [0,1] \tag{2}$$

$$q_{i,t} = q_{i,t} + g_i \odot \tilde{q}_{i,t} \in R^d \; q_{i,t} = \frac{q_{i,t}}{\parallel q_{i,t} \parallel} \tag{3}$$

When $V_k, V_q, V_Q \in A^{d\times d}$, σ represents the sigmoid function, and $\odot$ is the number product. When $V_k, V_q, V_Q \in A^{d\times d}$, σ represents the sigmoid function, and $\odot$ is the number product.

Because the user input Q_t may only be partially related to the problem condition, the gating function in Eq. (2) matches Q_t into the memory network. The Explicit Memory module then updates the revealed hidden state $v_{i,t}$ in a gated fashion. After reading the utterance information provided by the user and completing the implicit semantic conditional reasoning process, the key of the dialog and the updated latest state are expressed as $(k_1, q_1), \ldots, (k_M, q_M)$, which is then used for the inference decision module and implied intention prediction module.

6 Inference Decision

The final update key-value state based on the utterance question is expressed as $(k_1, q_1), \ldots, (k_M, q_M)$, the Inference Decision module predicts the decision in the Inquire module. Self-attention is used to compute a summary vector c of the overall intent and slot semantic state:

$$\omega_i = w_\omega^T[k_i; q_i] + b_\omega \in R^1 \tag{4}$$

$$\tilde{\omega}_i = \text{softmax}(\omega)_i \in [0, 1] \tag{5}$$

$$c = \sum_i \tilde{\omega}_i[k_i; q_i] \in R^d \tag{6}$$

The concatenation between the vectors k_1 and q_1 is represented as $[k_1; q_1]$, and ω_i is the attention weight of the sentence k_i in the dialog, which reflects the possibility of k_i in the dialog information provided by the user. Finally, a linear transformation of the vector c of the overall intent and the semantic state of the slot is performed:

$$z = W_z c + b_z \in A^4 \tag{7}$$

where the attention score $z \in A^4$ l indicates that the correct decision is selected, and formula (8) is used as the loss function to train the inference model:

$$\ell_{\text{dec}} = -\log \text{softmax}(z)_l \tag{8}$$

To determine whether the dialog intent has been fulfilled, a subtask is added to predict the semantically implicit state between the intent label and the slot label for each sentence. Semantic latent states include "implicit," "contradictory," and "unknown," and through this supervision mechanism, the model can make better decisions based on the implicit state of each sentence. Implicit inference prediction is taken from the Explicit Memory module by updating the final key-value state $[k_i; q_i]$.

$$v_i = V_v[k_i; q_i] + b_v \in A^3 \tag{9}$$

For the i-th dialog sentence, $v_i \in A^3$ has three implicit semantic states: $[\varphi_{\text{connotative},i}, \varphi_{\text{inconsistence},i}, \varphi_{\text{unknown},i}]$ scores, where r represents implicit. Thus, the semantic state contained in it is effectively predicted. Formula (10) trains the model as a loss function:

$$\ell_{\text{connota}} = -\frac{1}{M}\sum_{i=1}^{M} \log \text{softmax}(v_i)_r \tag{10}$$

7 Implied Intention Prediction

When the Inference Decision module infers an implicit semantic relationship, intention predictions are performed on the implicit relationship. This problem is decomposed into two stages. First, the segments of the conversation that contain implicit information about unknown users are extracted. Then, the unknown segment is extracted to predict the intent. A coarse-to-fine approach is employed to extract the unknown segments in the first stage.

A coarse-to-fine reasoning method is used to directly find the segments of unknown implicit relations without the complicated extraction of multiple segments from the dialog. Therefore, implicit semantic reasoning is used to predict the unknown scores $\varphi_{\text{unknown},i}$ in the subtask and to normalize them (in the dialog sentences) to determine if the i-th sentence contains the hidden information segment of the unknown utterance:

$$\zeta_i = \text{softmax}(\varphi_{\text{unknown}})_i \in [0, 1] \tag{11}$$

Understanding the likelihood of unknown implicit relationship segments in conversations greatly reduces the difficulty in extracting unknown utterance information segments. The user implicit relationship segment extraction (i.e., the start and endpoints of the predicted segment) score is adjusted using the dialog recognition score ζ_i by applying a soft selection method. The BERTQA method [16] is followed to build a start vector $w_s \in A^d$ and an end vector $w_v \in A^d$ in the segment to find the start and end positions from the whole conversation. The probability of the j-th word in the i-th sentence $u_{i,j}$ as the start/end of the implicit relationship segment is calculated as the dot product between w_s and $u_{i,j}$, where the in-dialog recognition score ζ_i is adjusted:

$$\gamma_{i,j} = w_s^T a_{i,j} * \zeta_i,\ \ \delta_{i,j} = w_v^T a_{i,j} * \zeta_i \tag{12}$$

In the case where the start and end positions must belong to the same conversation, the implicit relationship segment $\gamma * \delta$ with the highest score is extracted. Let s and v be the start and end positions, respectively, of the utterance implicit information segment. If the implicit relationship segment is not satisfied, the segment extraction loss is calculated as the pointing segment loss:

$$\ell_{sp,s} = -1_{l=inq} \log \text{softmax}(\gamma)_s \tag{13}$$

$$\ell_{sp,v} = -1_{l=inq} \log \text{softmax}(\delta)_v \tag{14}$$

The overall loss is the sum of the inference decision loss, the implicit semantic prediction loss, and the implicit relationship segment extraction loss:

$$\ell = \ell_{\text{dec}} + \lambda_1 \ell_{\text{connota}} + \lambda_2 \ell_{\text{sp}} \tag{15}$$

where λ_1 and λ_2 are tunable hyperparameters.

8 Experiment

8.1 Datasets

To evaluate the effectiveness of the proposed model, we conducted experiments on two public datasets, SNIPS [18]and ATIS [19]; the details of the datasets are presented in Table 1.

Table 1. Dataset information statistics

Dataset	Snips	ATIS
Vocab Size	11,241	722
Average Sentence Length	9.05	11.28
#Intent	7	21
#Slots	72	120
#Training Samples	13,084	4,478
#Validation Samples	700	500
#Test Samples	700	893

8.2 Baselines

To verify the effectiveness of the proposed model, the model was compared with the following traditional models:

Join sequence: [2] Used a bidirectional RNN to establish a joint model for slot filling, intent detection, and domain classification, supporting multi-task DL and realizing an end-to-end spoken language understanding framework.

Attention BiRNN: [3] Introduced the attention mechanism to their RNN and developed an encoding model for joint slot filling and intent detection, providing additional information for slot filling and intent detection.

Slot-gated full attention model: [4] Employed a slot-gate mechanism as a special gate function in an LSTM model to globally optimize the semantic framework results by learning the relationship between intent detection and slot filling attention vectors.

Joint capsule: [15] Adopted a capsule-based neural network model, utilized a dynamic routing protocol pattern for intent detection and slot filling.

A novel bidirectional association model: [20] Introduced the SF-ID network to establish a bidirectional association model for intent detection and slot filling.

Bert: [21] Used the BERT pretraining model for pretraining deep bidirectional representations of large-scale unlabeled corpora to address the problem of the poor generalization ability of spoken language comprehension models.[1]

BART: [22] A denoising automatic encoder for pre-training sequence-to-sequence models. BART reconstructs the original text by destroying it with an arbitrary noise function, learning a model.

8.3 Experimental Details

PyTorch was used to implement the proposed model. The model was trained sequentially in a pipelined manner. The advanced optimizer proposed by Adam [23] was used for training, with a learning rate of $3e^5$ and a warm-up rate of 0.1. The loss weights λ_1 and λ_2 were set as 10 and 0.6, respectively.

[1] https://github.com/jiangnanboy/intent_detection_and_slot_filling/tree/master/model6.

8.4 Experimental Results

The results showed that an explicit-memory few-shot joint learning model based on Siamese BERT outperforms other models in terms of intent recognition, slot filling, and overall sentence-level semantics. The comparisons are presented in Table 2. For Snips and ATIS datasets, Slot accuracy is increased by 0.3 and 0.4 respectively, while Intent accuracy is not significantly improved in Snips data sets, but increased by 0.1 in ATIS data sets. The accuracy of sentence-level semantic framework was increased by 0.1 and 0.3 respectively.

Table 2. Experimental results

Model	Snips			ATIS		
	Slot	Intent	Overall	Slot	Intent	Overall
Joint Seq	87.3	96.9	73.2	94.3	92.6	80.7
Attention BiRNN	87.8	96.7	74.1	94.2	91.1	78.9
Slot-Gated Full Atten	88.8	97.0	75.5	94.8	93.6	82.2
Joint Capsule	91.8	97.3	80.9	95.2	95.0	83.4
A Novel Bi-directional	92.2	97.3	80.4	95.6	96.6	86.0
Joint BERT	97.0	98.6	92.8	96.1	97.5	88.2
BART	97.2	98.8	93.0	96.3	97.7	88.1
Our model	97.5	98.8	93.1	95.7	97.8	88.4

8.5 Ablation Study

To study the effectiveness of the proposed model in slot filling and intent detection, we conducted ablation experiments on the SNIPS dataset to study the impact of each module in the model on the two tasks by replacing SBert with Bert and removing the Explicit Memory, Inference Decision, and Implied Intention Prediction modules. The ablation experiment results are presented in Table 3.

1. When we replaced SBert with Bert, the intent detection performance, the slot filling performance, and the overall performance decreased slightly; thus, the overall performance of SBert in spoken language comprehension is better than that of Bert.
2. When we removed the Explicit Memory module, the intent detection performance deteriorated slightly, the slot filling performance decreased reduced, and the overall performance deteriorated slightly; thus, indicating that the identification of implicit slots and intents is very beneficial.
3. When we removed the Inference Decision module, the performance deteriorated considerably, especially the performance of intent detection; thus, revealing that inference decisions help in predicting the semantic implications between the intent labels and slot labels of each sentence.

4. When we removed the Implied Intention Prediction module, the performance deteriorated notably because the implicit intent was not displayed, and the user's intent could not be truly understood, resulting in a substantial drop in the performance; thus, implicit intent prediction is helpful for intent detection and slot filling.

Table 3. Results of ablation experiments on the Snips dataset

Model	Snips		
	Slot	Intent	Overall
Our model	97.5	98.8	93.1
Our model(SBert → bert)	92.1	93.4	88.7
Our model(w/o Explict Memory)	92.3	95.6	89.1
Our model(w/o Inference Decision)	93.6	97.2	90.3
Our model(w/o Implied Intention Prediction)	91.2	92.3	87.4

8.6 Combined Pretraining

Recent studies have revealed that language model pretraining plays a vital role in many tasks (e.g., SBERT [16]). Inspired by the ease of scaling of language models through fine-tuning and the effectiveness of embeddings, we explored the joint experimental effect of our model + CRF. The results presented in Table 4 show the state-of-the-art performance of the proposed model on the SNIPS and ATIS datasets, It is shown that most of the parameters of our model are better than those of BERT model in the combined experiment on CRF.

Table 4. Results of joint experiments on Snips and ATIS, our model+CRF refers to the combination of our model and CRF.

Model	Snips			ATIS		
	Slot	Intent	Overall	Slot	Intent	Overall
Joint BERT+CRF	98.4	96.7	92.6	97.9	96.0	88.6
Our model	97.5	98.8	93.1	95.7	97.8	88.4
Our model+CRF	98.5	99.3	94.7	97.8	98.7	90.3

9 Conclusions and Future Work

In this study, we proposed an explicit-memory few-shot joint learning model based on Siamese BERT that adopts the similarity calculation method to obtain more intent and slot features to solve the problem of small-shot learning. Memory structures that explicitly track implicit semantic states between intents and slots in user-asked questions in conversations were employed. The updated intent and semantic state between slots were unified for the inference decision module, and the coarse-to-fine approach was employed for intent prediction. Experiments on the SNIPS and ATIS datasets demonstrated the effectiveness and generalization ability of the proposed model; moreover, the proposed method can be generalized for other types of spoken language understanding tasks.

References

1. Guo, D., Tur, G., Yih, W.-T., Zweig, G.: Joint semantic utterance classification and slot filling with recursive neural networks. In: 2014 IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, NV, pp. 554–559. IEEE (2014)
2. Hakkani-Tür, D., et al.: Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In: Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, pp. 715–719. ISCA (2016)
3. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, pp. 685–689. ISCA (2016)
4. Goo, C.-W., et al.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, vol. 2 (Short Papers), pp. 753–757. Association for Computational Linguistics (2018)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint (2014)
6. Liu, H., Zhang, F., Zhang, X., Zhao, S., Zhang, X.: An explicit-joint and supervised-contrastive learning framework for few-shot intent classification and slot filling. arXiv preprint (2021)
7. Kumar, M., Kumar, V., Glaude, H., de Lichy, C., Alok, A., Gupta, R.: Protoda: efficient transfer learning for few-shot intent classification. In: 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, pp. 966–972. IEEE (2021)
8. Zhang, J., et al.: Few-shot intent detection via contrastive pre-training and fine-tuning. In: 2021 EMNLP Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, pp. 1906–1912 (2021)
9. Su, J., Weijie, L., Yangyiwen, O.: Whitening sentence representations for better semantics and faster retrieval. arXiv preprint (2021)
10. Zhang, H., et al.: Fine-tuning pre-trained language models for few-shot intent detection: supervised pre-training and isotropization. arXiv preprint (2022)

11. Hashemi, H.B., Asiaee, A., Kraft, R.: Query intent detection using convolutional neural networks. In: International Conference on Web Search and Data Mining, Workshop on Query Understanding (2016)
12. Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D., Sarikaya, R.: Easy contextual intent prediction and slot detection. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, pp. 8337–8341. IEEE (2013)
13. Ravuri, S., Stolcke, A.: Recurrent neural network and LSTM models for lexical utterance classification. In: Interspeech 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, pp. 135–139. ISCA (2015)
14. Li, C., Li, L., Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 3824–3833. Association for Computational Linguistics (2018)
15. Zhang, C., Li, Y., Du, N., Fan, W., Yu, P.: Joint slot filling and intent detection via capsule neural networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 5259–5267. Association for Computational Linguistics (2019)
16. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using siamese BERT-networks. arXiv preprint (2019)
17. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates Inc. (2017)
18. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint (2018)
19. Tur, G., Hakkani-Tür, D., Heck, L.: What is left to be understood in ATIS? In: 2010 IEEE Spoken Language Technology Workshop, Berkeley, CA, pp. 19–24. IEEE (2010)
20. Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 5467–5471. Association for Computational Linguistics (2019)
21. Chen, Q., Zhuo, Z., Wang, W.: BERT for joint intent classification and slot filling. arXiv preprint (2019)
22. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint (2019)
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA (2015)

A Noise-Removal of Knowledge Graph Framework for Profile-Based Spoken Language Understanding

Leyi Lao, Peijie Huang(✉), Zhanbiao Zhu, Hanlin Liu, Peiyi Lian, and Yuhong Xu

College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
pjhuang@scau.edu.cn

Abstract. To alleviate the ambiguity of user utterances, a profile-based spoken language understanding (PROSLU) task has been proposed recently, which uses supporting profile information as supplementary knowledge. The knowledge graph, as one of the supporting profile information, contains many entities with rich attribute information which do help the performance of SLU. However, existing models treat all entities as useful, resulting in a lot of noise being fed. Moreover, not all the attribute information of the entity is necessary, and there is a lot of noise and redundancy. In this paper, we propose a Noise-Removal of Knowledge Graph framework for PROSLU (NRKG-PROSLU), with two different kinds of denoising. One is a hard denoising by entity selection, where we introduce a small clean dataset and propose an auxiliary model, BERT-based entity selection, to filter out entities irrelevant to user utterances, namely noisy entities. The other is a soft denoising by entity attribute information selection, where we introduce keywords-based local semantic selection, which uses keywords to give more weight to relevant local semantics, so as to capture task-related information in the selected entities for reducing noise and redundancy. Then, the denoised knowledge graph is used to assist the SLU model, which achieves better performance than competitive models on the public PROSLU dataset.

Keywords: Profile-based Spoken Language Understanding · Intent Detection · Slot Filling · Noise-Removal

1 Introduction

Spoken language understanding (SLU) is the core component of task-based dialogue systems, which aims to obtain the frame semantic representation information of user utterances [1–3]. It contains two subtasks, one is intent detection and the other is slot filling. State-of-the-art models use joint models to model the relation between them because of the strong correlation between the two subtasks. Some works [4–6] implicitly connect intent detection and slot filling by

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 814–826, 2023.
https://doi.org/10.1007/978-3-031-44693-1_63

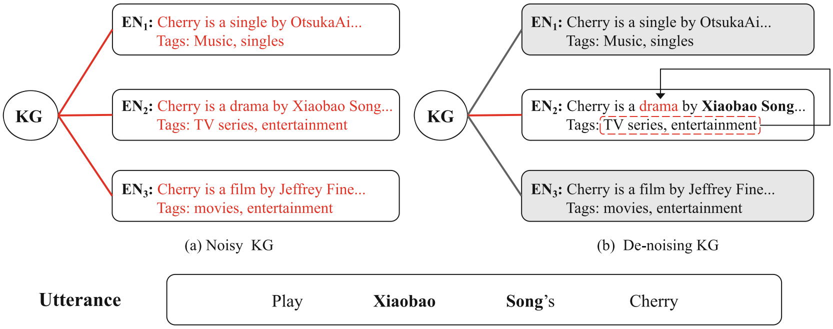

KG
EN1: Cherry is a single by OtsukaAi... Tags: Music, singles
EN2: Cherry is a drama by Xiaobao Song... Tags: TV series, entertainment
EN3: Cherry is a film by Jeffrey Fine... Tags: movies, entertainment
(a) Noisy KG
KG
EN1: Cherry is a single by OtsukaAi... Tags: Music, singles
EN2: Cherry is a drama by Xiaobao Song... Tags: TV series, entertainment
EN3: Cherry is a film by Jeffrey Fine... Tags: movies, entertainment
(b) De-noising KG
Utterance
Play
Xiaobao
Song's
Cherry

Fig. 1. An example of KG denoising. (a) No denoising is performed. (b) Remove irrelevant entities and give more weight to key local semantics.

sharing parameters. They outperform pipeline models by mutual enhancement between the two tasks. Recently, some joint models [7–12] utilize intent information to explicitly guide the slot filling task and achieve advanced performance. However, the studies of these works are limited to a simple setting, which is the plain text-based SLU.

In real-world scenarios, the user utterance often faces semantic ambiguity due to colloquial input. It is not enough to extract the correct semantic frame information if only relying on the user utterances. To this end, Xu et al. [13] proposed profile-based SLU (PROSLU), which adds the supporting profile information as supplementary knowledge to alleviate the ambiguity of user utterances. They adopted the hierarchical attention fusion mechanism [14,15] as a multi-level knowledge adapter to fuse three types of supporting profile information: (1) Knowledge Graph (KG), which consists of entities with rich attributes, (2) User Profile (UP), which consists of user information and settings, and (3) Context Awareness (CA), which is environmental information and user state. And it is compatible with the existing SLU model in the form of a plug-in.

Despite the promising performance achieved, the large number of entities and rich attribute information bring great challenges to the KG encoder. The existing models have the following two problems in processing KG, resulting in the inability to effectively capture important information:

- A large number of entities are fed into the model, but quite a few of them are not relevant to user utterances, namely noisy entities. As an example in Fig. 1(a), for the user utterance "Play Xiaobao Song's Cherry", in its KG, we can be known that the first and third entities are noisy and harmful, they cannot help the model to correctly predict the semantic frame, or even mislead.
- The attribute information of each entity is diverse, but not all of the information is required, that is, it usually contains noise or redundancy. Take the second entity in Fig. 1(b) for example, where "drama" is the key piece of information. Moreover, the sparsity and complexity of attribute information

aggravate the gradient disappearance problem, resulting in the loss of important local semantic information. However, they do not have a clear model to capture the key semantic information.

In this paper, to address the issues described above, we propose a Noise-Removal of Knowledge Graph framework for Profile-based SLU (NRKG-PROSLU), which provides the multi-level knowledge adapter with appropriate denoised KG, as shown in Fig. 1(b). For the first problem, we use a hard denoising method to alleviate the impact of noisy entities on the model. To this end, we introduce a small clean PROSLU++ dataset and train an auxiliary model, BERT-based entity selection. After only learning on the small clean dataset, the auxiliary model can denoise the rest of the noisy dataset, leaving entities relevant to user utterances and more effectively alleviating the ambiguity of user utterances. For the second problem, we introduce a keywords-based local semantic selection, which utilizes a soft denoising method to enable the model to focus on important information. According to the characteristics of entity attribute information, we directly extract the specific "Tags" part as keywords, as shown in Fig. 1(b). Keywords are then used to give more weight to key local semantics, so as to accurately capture information related to the task for reducing noise and redundancy. We perform experiments on the public PROSLU dataset, and the results show that our proposed framework is superior.

2 Related Work

Dominant SLU systems use the joint model to model the relationship between intent detection and slot filling. The existing joint models can be divided into two main categories. The first category of work [4–6] adopted a multi-task framework with a shared encoder to implicitly model the relationship between these two tasks. The second category of work utilized intent information to explicitly guide the slot filling task. It includes the Slot-Gated [7], the Bi-Model [8], the SF-ID [10], the Stack-Propagation [12], and so on. However, the research of these models mainly focuses on the SLU based on plain text. It is difficult to capture the correct semantic frame when the user utterance has semantic ambiguity.

Recently, based on the consideration of SLU in ambiguous settings, Xu et al. [13] first proposed a new important task, ProSLU, which adds the supporting profile information as supplementary knowledge to alleviate the ambiguity of user utterances. However, their works ignored whether KG entities are relevant to user utterances, and did not consider the sparsity and complexity of each entity attribute information. Compared with their works, our model performs two different denoising operations on KG information to further alleviate semantic ambiguity.

3 Proposed Model

Our study does not pick up the base model, which can be compatible with many existing SLU models. And our current model is based on General-PROSLU [13], and our main contribution is noise removal of KG, including entity selection and local semantic selection, as shown in Fig. 2.

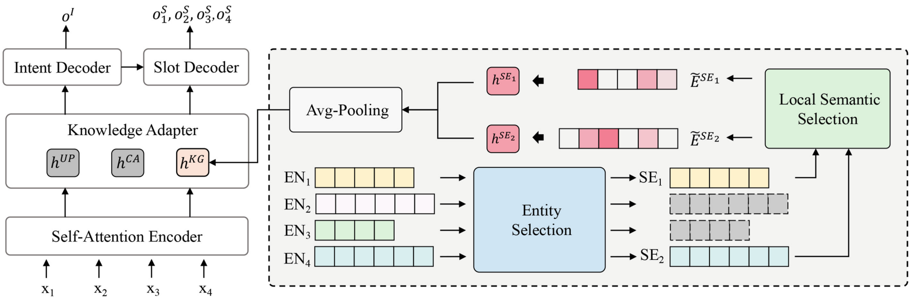


Fig. 2. Illustration of our proposed NRKG-PROSLU.

3.1 Self-attention Encoder

Given a sequence of user utterance containing T tokens $\{t_1, t_2, ..., t_T\}$, it is fed to the embedding layer ϕ^{emb} and thus mapped into an embedding sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\} \in \mathbb{R}^{T \times d}$ (d represents the dimension of the embedding). The BiLSTM [16] with self-attention mechanism can well utilize the advantages of word order features and contextual information. Following Xu et al. [13], we use BiLSTM to generate the context-sensitive hidden states $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_T\}$, where $\boldsymbol{h_i} = \text{BiLSTM}(\boldsymbol{x}_i)$. We use the self-attention mechanism for the token representation matrix, which is inspired by Vaswani et al. [17]: $\boldsymbol{C} = \text{Self-Attention}(\boldsymbol{X})$. We concatenate $\boldsymbol{H}$ and $\boldsymbol{C}$ into a matrix $\boldsymbol{E} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_T\} \in \mathbb{R}^{T \times 2d}$ as follows:

$$\boldsymbol{E} = \boldsymbol{H} \oplus \boldsymbol{C}, \tag{1}$$

where $\oplus$ represents concatenation.

3.2 Knowledge Adapter

Following Xu et al. [13], there are three types of supporting profile information: UP, CA, and KG. For UP and CA, given their single feature vector $\boldsymbol{x}_{UP} \in \mathbb{R}^u$ and $\boldsymbol{x}_{CA} \in \mathbb{R}^c$ (u and c are the UP and CA feature dimensions respectively). Representations of UP and CA can be obtained by linear projection, using $\boldsymbol{h}^{UP} = \boldsymbol{W}_{UP}^{\top} \boldsymbol{x}_{UP}$ and $\boldsymbol{h}^{CA} = \boldsymbol{W}_{CA}^{\top} \boldsymbol{x}_{CA}$, where $\boldsymbol{W}_{UP} \in \mathbb{R}^{u \times d_s}$ and $\boldsymbol{W}_{CA} \in \mathbb{R}^{c \times d_s}$ are trainable parameters and d_s is the embedding dimension.

For KG with N entities $KG = \{EN_1, ..., EN_N\}$, we need to select M entities associated with the user utterance:

$$\{SE_1, ..., SE_M\} = \text{Entity-Selection}(KG). \tag{2}$$

For one of the selected entities SE_i, the weighted representation is obtained by keywords-based local semantic selection:

$$\tilde{\boldsymbol{E}}^{SE_i} = \text{LocalSemantic-Selection}(SE_i), \tag{3}$$

where $\tilde{\boldsymbol{E}}^{SE_i} = \{\tilde{\boldsymbol{e}}_1, \tilde{\boldsymbol{e}}_2, ..., \tilde{\boldsymbol{e}}_n\}$ (n is the number of tokens in the input entity). Score each element of the sequence with a linear layer and obtain the final representation of the entity:

$$\boldsymbol{h}^{SE_i} = \boldsymbol{W}_h \sum_{j=1}^{n} \frac{\exp(\boldsymbol{W}_e \tilde{\boldsymbol{e}}_j)}{\sum_{k=1}^{n} \exp(\boldsymbol{W}_e \tilde{\boldsymbol{e}}_k)} \odot \tilde{\boldsymbol{e}}_j, \tag{4}$$

where $\odot$ is element-wise multiplication, $\boldsymbol{W}_e$ and $\boldsymbol{W}_h$ are trainable parameters.

For multiple selected entities, we perform average pooling on their final representation as the overall aggregated selected entities representation:

$$\boldsymbol{h}^{KG} = \text{Avg-Pooling}(\boldsymbol{h}^{SE_1}, ..., \boldsymbol{h}^{SE_M}). \tag{5}$$

Following Xu et al. [13], we adopt the hierarchical attention fusion mechanism as the knowledge adapter to fuse supporting profile information $\boldsymbol{H}^{Info} = [\boldsymbol{h}^{UP}; \boldsymbol{h}^{CA}; \boldsymbol{h}^{KG}] \in \mathbb{R}^{3 \times d_s}$. Given the query vector $\boldsymbol{q}$, we can obtain the updated representation $\boldsymbol{q'} = \text{Knowledge-Adapter}(\boldsymbol{q}, \boldsymbol{H}^{Info})$ as follows:

$$\boldsymbol{q'} = \sum_{i=1}^{3} \frac{\exp(\boldsymbol{q}\boldsymbol{W}\boldsymbol{h}_i^{info})}{\sum_{k=1}^{3} \exp(\boldsymbol{q}\boldsymbol{W}\boldsymbol{h}_k^{info})} \boldsymbol{h}_i^{info}, \tag{6}$$

where W is trainable parameter and $\{\boldsymbol{h}_1^{info}; \boldsymbol{h}_2^{info}; \boldsymbol{h}_3^{info}\}$ denotes $\{\boldsymbol{h}^{UP}; \boldsymbol{h}^{CA}; \boldsymbol{h}^{KG}\}$ respectively.

3.3 Entity Selection

Entity selection aims to remove noisy entities not related to user utterance, so as to achieve hard denoising. Xu et al. [13] utilized the supporting profile information as supplementary knowledge to alleviate the ambiguity of user utterances, but they ignored the relevance between the entities of the KG and the user utterances. The input of a large number of noisy entities may lead to sub-optimal performance. Hence, we introduce an auxiliary model to remove noisy entities.

First, we label a small clean dataset, as shown on the left of Fig. 3, and train an auxiliary model on it. Then, the trained model is used to predict the correct ***KG_labels*** for each sample in the noisy dataset, as shown on the right of Fig. 3. Specifically, given the user utterance U and its $KG = \{EN_1, ..., EN_N\}$, we concatenate them separately with a special $[SEP]$ token, resulting in N sentence pairs:

$$I^{EN_i} = U \oplus [SEP] \oplus EN_i. \tag{7}$$

We utilize BERT [18], which is a pre-trained transformer network [17], to encode sentence pairs into contextual semantic representations $\{\boldsymbol{H}^{UE_1}, ..., \boldsymbol{H}^{UE_N}\}$ as follows:

$$\boldsymbol{H}^{UE_i} = \text{BERT}(I^{EN_i}), \tag{8}$$

where $\boldsymbol{H}^{UE_i} \in \mathbb{R}^{|I^{EN_i}| \times d_B}$. Here, $|I^{EN_i}|$ and d_B denote the sequence length of I^{EN_i} and the BERT output dimension, respectively. Then, the decoder reads them to obtain the $\boldsymbol{KG\_labels} \in \mathbb{R}^N$, where the ones labeled as 1 are all the selected entities $\{SE_1, ..., SE_M\}$, and the other ones are noisy entities:

$$\boldsymbol{KG\_labels} = \text{Decoder}(\boldsymbol{H}^{UE_1}, ..., \boldsymbol{H}^{UE_N}). \tag{9}$$

Finally, the noise entities were eliminated to reduce the influence of a large amount of noise on the main model.

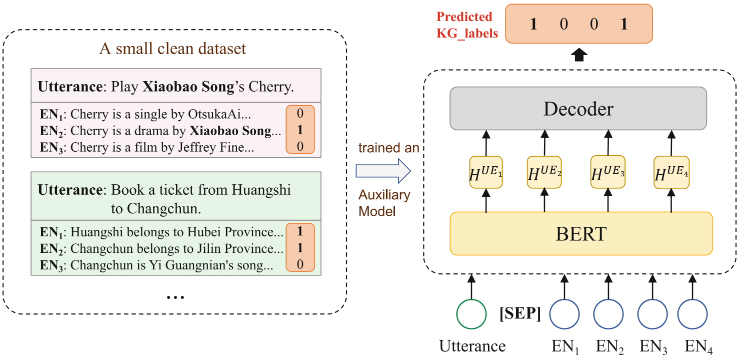

A small clean dataset
Utterance: Play Xiaobao Song's Cherry.
EN1: Cherry is a single by OtsukaAi...
EN2: Cherry is a drama by Xiaobao Song...
EN3: Cherry is a film by Jeffrey Fine...
0
1
0
Utterance: Book a ticket from Huangshi to Changchun.
EN1: Huangshi belongs to Hubei Province...
EN2: Changchun belongs to Jilin Province...
EN3: Changchun is Yi Guangnian's song...
1
1
0
...
trained an
Auxiliary Model
Predicted KG_labels
1 0 0 1
Decoder
H^{UE_1}
H^{UE_2}
H^{UE_3}
H^{UE_4}
BERT
[SEP]
Utterance
EN1
EN2
EN3
EN4

Fig. 3. BERT-based entity selection.

3.4 Local Semantic Selection

The attribute information of entities is often very diverse and rich, but at the same time there is a lot of noise and redundancy. For example, in the attribute information“The Great Star is a song performed by Jay Chou and is an insert song in the movie Rooftop. The release date is July 8, 2013. Tags: music”, the word “movie” is noisy. In addition, much of the attribute information is useless and redundant. Therefore, the sparsity and complexity of entity attribute information pose great challenges to encoders.

The “Tags” section of the original entity attribute information contains more condensed semantics and is a key description of the information. Therefore, we treat the “Tags” section as keywords and use them to build a selection gate. Here, the information of keywords is a powerful selection signal, which aims

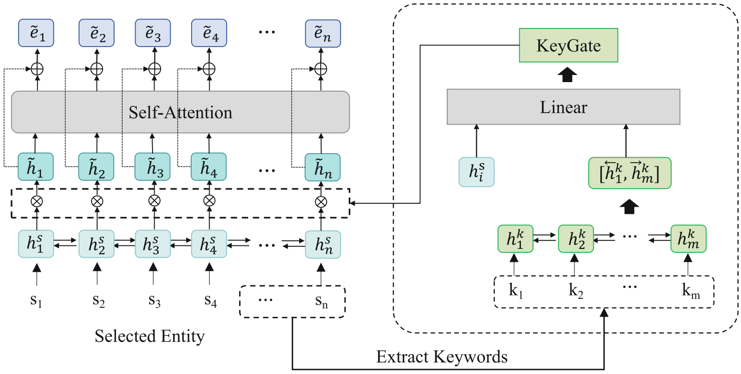


Fig. 4. Keywords-based local semantic selection.

to assign more weight to the key local semantics, so as to accurately capture task-relevant information, as shown in Fig. 4.

For the selected entities SE_i, its embedding sequence $\boldsymbol{S} = \{s_1, s_2, ..., s_n\} \in \mathbb{R}^{n \times d_s}$ can obtain the context-sensitive hidden states $\boldsymbol{H}^s = \{\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, ..., \boldsymbol{h}_n^s\}$ through $\boldsymbol{h}_i^s = \text{BiLSTM}(s_i)$. The same operation is used to get the keywords context-sensitive hidden states $\boldsymbol{H}^k = \{\boldsymbol{h}_1^k, \boldsymbol{h}_2^k, ..., \boldsymbol{h}_m^k\}$, thus enabling us to obtain the keywords representation $\boldsymbol{a}^k = [\overleftarrow{\boldsymbol{h}_1^k}; \overrightarrow{\boldsymbol{h}_m^k}]$, where m is the number of tokens in the keywords. We use $\boldsymbol{a}^k$ to obtain the keywords-selective gate, which guides the encoding of the original entity attribute information, thus obtaining the updated representation $\tilde{\boldsymbol{H}} = \{\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, ..., \tilde{\boldsymbol{h}}_n\}$ as follows:

$$\boldsymbol{keyGate}_i = \sigma(\boldsymbol{W}_k \boldsymbol{h}_i^s + \boldsymbol{U}_k \boldsymbol{a}^k), \tag{10}$$

$$\tilde{\boldsymbol{h}}_i = \boldsymbol{keyGate}_i \odot \boldsymbol{h}_i^s, \tag{11}$$

where σ denotes the sigmoid function, $\boldsymbol{W}_k$ and $\boldsymbol{U}_k$ are trainable parameters. We perform self-attention on $\tilde{\boldsymbol{H}}$, which gives higher weight to key local semantics depending on the context. We concatenate the output and $\tilde{\boldsymbol{H}}$ to get a weighted representation $\tilde{\boldsymbol{E}}^{\boldsymbol{SE_i}} \in \mathbb{R}^{n \times 2d_s}$ of the selected entity SE_i:

$$\tilde{\boldsymbol{C}} = \text{Self-Attention}(\tilde{\boldsymbol{H}}), \tag{12}$$

$$\tilde{\boldsymbol{E}}^{SE_i} = \tilde{\boldsymbol{H}} \oplus \tilde{\boldsymbol{C}}. \tag{13}$$

3.5 Intent Detection and Slot Filling Decoder

The user utterance representation $\boldsymbol{g}$ can be generated on the shared encoding representation $\boldsymbol{E}$. Following Xu et al. [13], we use the sentence-level knowledge adapter to obtain $\boldsymbol{f}^{info} = \text{Knowledge-Adapter}(\boldsymbol{g}, \boldsymbol{H}^{Info})$ for enhancing intent detection:

$$o^I = \text{argmax}(\text{softmax}(\boldsymbol{W}_I(\boldsymbol{g} \oplus \boldsymbol{f}^{info}))). \tag{14}$$

Following Xu et al. [13], the self-attention encoding e_t at the t-th timestep is used as the query vector. We use the word-level knowledge adapter to obtain $\boldsymbol{f}_t^{info} = \text{Knowledge-Adapter}(\boldsymbol{e}_t, \boldsymbol{H}^{Info})$, which is used for augmenting the word-level representation in the slot filling decoder:

$$\boldsymbol{h}_t^S = \text{LSTM}(\boldsymbol{s}_t \oplus \boldsymbol{f}_t^{info}, \boldsymbol{h}_{t-1}^S), \tag{15}$$

$$o_t^S = \text{argmax}(\text{softmax}(\boldsymbol{W}_S \boldsymbol{h}_t^S)), \tag{16}$$

where $\boldsymbol{s}_t$ is the concatenation of the previous slot embedding, the predicted intent embedding and the aligned encoder hidden state $\boldsymbol{e}_t$. And $\boldsymbol{W}_I$, $\boldsymbol{W}_S$ are trainable parameters.

4 Experiment

4.1 Experiment Setting

Datasets. The proposed framework is evaluated on the public PROSLU [13] dataset, which consists of 4196, 522, and 531 user utterances for training, validation, and testing, respectively. To obtain a small clean PROSLU++ dataset, 500 and 50 user utterances are randomly selected from the noisy training set and noisy validation set of [13], respectively, to manually label their KG. The PROSLU++ dataset is used to train the auxiliary model to predict the correct KG labels for other noisy datasets.

Implementation Details. The embedding size is set to 64. The self-attentive encoder hidden states are set to 256. For reducing overfitting, the dropout rate is 0.4. Adam [19] is used to optimize the model parameters and the suggested hyper-parameters are used for optimization. Experiment results are the average of 5 independent experiments.

Baselines. We compare our NRKG-PROSLU with the following existing state-of-the-art SLU models, all of which are compatible with multi-level knowledge adapters: (1) Slot-Gated [7], which explicitly models the correlation between intent detection and slot filling. (2) Bi-Model [8], which studies the cross influence between slot filling and intent detection. (3) SF-ID [10] network for constructing direct connections of slot filling and intent detection. (4) A joint model with Stack-Propagation [12], which can capture the intent semantic information. (5) General-PROSLU proposed by Xu et al. [13], which is the basis of our model.

4.2 Overall Results

Following Goo et al. [7] and Xu et al. [13], we use Slot(F1) and Intent(Acc) to evaluate the performance of slot filling and intent detection, respectively, and use Overall(Acc) to evaluate sentence-level semantic framework, which shows that both slot filling and intent detection are correct. The results for the comparison models in Table 1 are taken from [13].

We draw the following conclusions based on the results in Table 1: (1) NRKG-PROSLU achieves the best performance on all metrics across all baselines. (2) Our NRKG-PROSLU outperforms the optimal scores in the baselines by 6.94% on Slot(F1), 6.93% on Intent(Acc), and 5.42% on Overall(Acc). This proves that NRKG-PROSLU has superior performance in filtering and utilizing KG, so as to alleviate the ambiguity of utterances more effectively and improve the performance of SLU models in obtaining the semantic frame.

Table 1. Slot Filling and Intent Detection results on the PROSLU dataset.

Model	Slot(F1)	Intent(Acc)	Overall(Acc)
Slot-Gated	74.18	83.24	69.11
Bi-Model	77.76	82.30	73.45
SF-ID	73.70	83.24	68.36
Stack-Propagation	81.08	83.99	78.91
General-PROSLU	83.27	85.31	79.10
NRKG-PROSLU	**90.21**	**92.24**	**84.52**

Table 2. Performance of pre-trained-Based PROSLU models.

Model	General-PROSLU			NRKG-PROSLU		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
RoBERTa	82.90	85.31	81.17	90.25	91.49	86.16
ELECTRA	84.38	86.63	82.30	**92.23**	**93.07**	**88.47**

4.3 Performance of Pre-trained-based NRKG-PROSLU Models

We investigate the performance of pre-trained-based NRKG-PROSLU models. Following Xu et al. [13], we adopt the pre-trained models RoBERTa [20] and ELECTRA [21] as the shared encoder, thus obtaining the pre-trained-based NRKG-PROSLU models. The results of the pre-trained-based General-PROSLU models are taken from [13]. As shown in Table 2, based on RoBERTa, our model outperforms General-PROSLU by 7.35% on Slot(F1), 6.18% on Intent(Acc), and 4.99% on Overall(Acc). Besides, based on ELECTRA, we achieve 7.85%, 6.44%, and 6.17% improvements on Slot(F1), Intent(Acc), and Overall(Acc). This shows that NRKG-PROSLU can be well-compatible with different pre-trained models.

4.4 Ablation Study of NRKG-PROSLU

In order to study the effectiveness of different modules of NRKG-PROSLU, the best model in Table 2, ELECTRA-based NRKG-PROSLU model, is studied by ablation, as shown in Table 3. First, we experimented with only BERT-based entity selection and observed large improvements of 7.05%, 5.95%, and 5.57% on Slot(F1), Intent(Acc), and Overall(Acc). This shows that it can reduce a large number of noisy entities entering the model and effectively alleviate the ambiguity of user utterances. Then, we add keywords-based local semantic selection on this basis. We found further improvements of 0.80%, 0.49%, and 0.60% on the three metrics. It can be seen that this module alleviates the local semantic loss and further improves the SLU performance of the model.

Table 3. Ablation study of NRKG-PROSLU.

Model	Slot(F1)	Intent(Acc)	Overall(Acc)
General-PROSLU	84.38	86.63	82.30
+Entity Selection	91.43	92.58	87.87
+Entity Selection & Local Semantic Selection	**92.23**	**93.07**	**88.47**

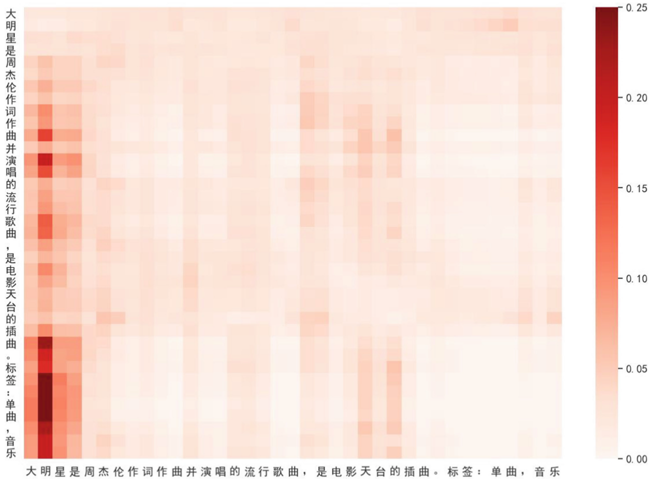


Fig. 5. Visualization of local semantic selection.

4.5 Visualization of Local Semantic Selection

To better understand how local semantic selection affects and selects key information, we visualize it as shown in Fig. 5. We can clearly observe that after the keywords-based local semantic selection operation, the key local semantic information in the text is given more weight, so that they gain more attention. This experiment shows that our method can let the model capture the key information in the entity attribute information more accurately, so as to assist the SLU model to predict the correct intent and slot more effectively.

4.6 Category Benefit Analysis

Based on intent, the PROSLU dataset can be classified into ten categories: music, voice, video, coach, bus, drive, flight, train, location, and metro. To see which categories benefit more from our model, we compare our NRKG-PROSLU with General-PROSLU. Table 4 shows five categories that benefit the most. We run the published General-PROSLU code and the results are labeled with †. We observe that the categories of music, voice, and video, which belong to entertainment, gain improvements of more than 15% in all metrics. While bus and metro, which belong to transportation, have smaller improvements. Notice that there is more KG noise in the entertainment categories compared to the transportation categories. This demonstrates that the improvements gained of our model are much more significant for the categories that with noisier KG.

Table 4. The top five categories that benefit the most.

Category	General-PROSLU †			NRKG-PROSLU		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Music	64.52	65.54	57.85	$83.01_{\uparrow 18.5}$	$86.46_{\uparrow 20.9}$	$73.54_{\uparrow 15.7}$
Voice	62.67	65.63	56.26	$87.13_{\uparrow 24.5}$	$90.94_{\uparrow 25.3}$	$77.82_{\uparrow 21.6}$
Video	58.57	56.62	50.15	$77.58_{\uparrow 19.0}$	$78.77_{\uparrow 22.2}$	$65.54_{\uparrow 15.4}$
Train	93.21	91.91	90.48	$95.71_{\uparrow 2.5}$	$94.29_{\uparrow 2.38}$	$93.81_{\uparrow 3.3}$
Metro	89.88	92.56	85.12	$92.70_{\uparrow 2.8}$	$96.74_{\uparrow 4.2}$	$87.44_{\uparrow 2.3}$

5 Conclusion

In this paper, we propose a Noise-Removal of Knowledge Graph framework for Profile-based Spoken Language Understanding (NRKG-PROSLU), which performs noise removal on two modules: entity selection and local semantic selection, thereby more effectively alleviating the ambiguity of user utterances. Experimental results on the public PROSLU dataset show that our framework is superior to the state-of-the-art methods, especially for the categories that with noisier KG. In addition, we experimentally demonstrate that the proposed NRKG-PROSLU model can be well-compatible with different pre-trained models. And we use ablation studies to confirm the effectiveness of each module.

Acknowledgements. This work was supported by Natural Science Foundation of Guangdong Province (No. 2021A1515011864) and Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation (No. pdjh2022a007).

References

1. Tur, G., De Mori, R.: Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. Wiley, New York (2011)
2. Yao, K., Peng, B., Zweig, G., Yu, D., Li, X., Gao, F.: Recurrent conditional random field for language understanding. In: Proceedings of ICASSP 2014, Florence, Italy, 4–9 May 2014, pp. 4077–4081 (2014)
3. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: Proceedings of SLT 2014, South Lake Tahoe, NV, USA, 7–10 December 2014, pp. 189–194 (2014)
4. Liu, B., Lane, I.R.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Proceedings of INTERSPEECH 2016, San Francisco, CA, USA, 8–12 September 2016, pp. 685–689 (2016)
5. Hakkani-Tür, D.: Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In: Proceedings of INTERSPEECH 2016, San Francisco, CA, USA, 8–12 September 2016, pp. 715–719 (2016)
6. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: Proceedings of IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 2993–2999 (2016)
7. Goo, C.-W.: Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, pp. 753–757 (2018)
8. Wang, Y., Shen, Y., Jin, H.: A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In: Proceedings of NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, pp. 309–314 (2018)
9. Li, C., Li, L., Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of EMNLP 2018, Brussels, Belgium, 31 October–4 November 2018, pp. 3824–3833 (2018)
10. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of ACL 2019, Florence, Italy, 28 July–2 August 2019, pp. 5467–5471 (2019)
11. Zhu, Z., Huang, P., Huang, H., Liu, S., Lao, L.: A graph attention interactive refine framework with contextual regularization for jointing intent detection and slot filling. In: Proceedings of ICASSP 2022, Virtual and Singapore, 23–27 May 2022, pp. 7617–7621 (2022)
12. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proceedings of EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 2078–2087 (2019)
13. Xu, X., Qin, L., Chen, K., Wu, G., Li, L., Che, W.: Text is no more enough! A benchmark for profile-based spoken language understanding. In: Proceedings of AAAI 2022, Vancouver, BC, Canada, 22 February–1 March 2022, pp. 11575–11585 (2022)

14. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1412–1421 (2015)
15. Libovický, J., Helcl, J.: Attention strategies for multi-source sequence-to-sequence learning. In: Proceedings of ACL 2017, Vancouver, Canada, 30 July–4 August 2017, pp. 196–202 (2017)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
17. Vaswani, A.: Attention is all you need. In: Proceedings of NIPS 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 5998–6008 (2017)
18. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, pp. 4171–4186 (2019)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of ICLR 2015, San Diego, CA, USA, 7–9 May 2015
20. Liu, Y.: RoBERTa: a robustly optimized BERT pretraining approach. CoRR, abs/1907.11692 (2019)
21. Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: Proceedings of ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020

Bilevel Scheduled Sampling for Dialogue Generation

Jiawen Liu and Kan Li(✉)

Beijing Institute of Technology, Beijing, China
{liujiawen,likan}@bit.edu.cn

Abstract. Exposure bias poses a common challenge in numerous natural language processing tasks, particularly in the dialog generation. In response to this issue, researchers have devised various techniques, among which scheduled sampling has proven to be an effective method for mitigating exposure bias. However, the existing state-of-the-art scheduled sampling methods solely consider the current sampling words' quality for threshold truncation sampling, which overlooks the importance of sentence-level information and the method of threshold truncation warrants further discussion. In this paper, we propose a bilevel scheduled sampling model that takes the sentence-level information into account and incorporates it with word-level quality. To enhance sampling diversity and improve the model's adaptability, we propose a smooth function that maps the combined result of sentence-level and word-level information to an appropriate range, and employ probabilistic sampling based on the mapped values instead of threshold truncation. Experiments conducted on the DailyDialog and PersonaChat datasets demonstrate the effectiveness of our proposed methods, which significantly alleviate the exposure bias problem and outperform state-of-the-art scheduled sampling methods.

Keywords: Scheduled Sampling · Exposure Bias · Dialog Generation

1 Introduction

Exposure bias is a common problem in many natural language processing tasks [1], especially in dialog generation. Exposure bias refers to the discrepancy between the training and inference stages of a model, where the model is trained on ground truth data but generates on its own predictions at inference time. This results in the model facing different environments during training and inference. Once a model prediction is inconsistent with the ground truth somewhere, it can affect later predictions, which will lead to errors accumulating and propagating along the generated sequence, resulting in poor quality, diversity and generalization of responses.

To address this issue, researchers have developed various techniques, such as beam search [2], DAD(Data As Demonstrator) [3], sentence level training [4],

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 827–839, 2023.
https://doi.org/10.1007/978-3-031-44693-1_64

scheduled sampling [1,5], reinforcement learning explicitly trains models [6,7], etc. These techniques help models handle long-term dependencies better in inference process. Currently, scheduled sampling is widely adopted and has demonstrated favorable outcomes. Nevertheless, the existing methods either generate entire sentences autoregressively before sampling [8], or solely consider the current sampling word's impact for threshold truncation sampling [9,10]. The former necessitates multiple beam searches, resulting in high computational complexity, low efficiency, and cannot dynamically adjust based on the quality of each word, thereby overlooking the differences between words in the sentence. The latter disregards sentence-level information and the method of threshold truncation warrants further discussion.

In our opinion, the importance of word-level information is obvious, and current methods that sample based on word-level scores [9,10] are significantly better than traditional methods [1,8] that sample each word in a sentence with the same probability. However, relying on cosine similarity or other scores between the generated word and the ground truth word is limited, and we still need to take the sentence-level quality into account. Therefore, in this paper, we propose a bilevel scheduled sampling model that dynamically adjusts the sampling probabilities of the generated results and the ground truth at both word and sentence level. Our model effectively balances the trade-off between learning from the ground truth and its own predictions, thereby enhancing the robustness and diversity of dialogue generation.

Specifically, sentence quality evaluation in sentence-level sampling is conducted through the utilization of either BLEU score or sentence-level cosine similarity. On the other hand, word quality in word-level sampling is evaluated based on the probability of word generation. Finally, a specially designed smooth function is used to integrate sentence-level and word-level evaluations, enabling probabilistic sampling.

We evaluate the proposed model on two publicly available datasets, namely DailyDialog and PersonaChat. Experimental results show that our model is superior to existing exposure bias models in both metric-based automatic evaluation and human evaluation. The main contributions of this paper include:

- To the best of our knowledge, we are the first to propose a bilevel scheduled sampling model that takes both sentence-level and word-level information into account.
- In sentence-level sampling, we utilize BLEU and sentence-level cosine similarity as evaluation metrics. For word-level sampling, we leverage the predicted probabilities as the word-level score. To enhance the diversity of the sampling process and improve the model's adaptability, we propose a smoothing function that maps the combined results of bilevel sampling to an appropriate range and adopt probabilistic sampling instead of threshold truncation.
- Extensive evaluations on two widely used open-domain dialogue datasets demonstrate that the proposed approach significantly alleviates the exposure bias problem and outperforms the state-of-the-art scheduled sampling methods.

2 Related Work

Data As Demonstrator (DAD) is a meta learning algorithm [3] that solves the problem of exposure bias by blending real training data with model predictions. During the training phase, not only the ground truth is used as input, but also the model-predicted results are added to the training set to make it conform to the test distribution. The Scheduled Sampling method [1] further developed this approach for sequence generation by sampling from the model's own predictions during training, instead of always using the ground truth. This method was first proposed by Bengio et al. in 2015 [1], and was later improved for transformer models by Mihaylova and Martins in 2019 [5], owing to the superior performance of transformer models [11]. What's more, since scheduled sampling only looks ahead one step, Goodman et al. proposed the TeaForN method [12], which looks ahead N steps for more foresight in sampling, but at the cost of reduced training efficiency.

The initial strategy for sampling was to decay the probability of sampling the ground truth based on training steps, with specific linear decay $f(i) = max(\epsilon, ki + b)$, exponential decay $f(i) = k^i$, and sigmoidal decay $f(i) = \frac{k}{k+e^{\frac{i}{k}}}$. Later in 2021, Liu et al. proposed a strategy of decaying according to decoding steps [13], further improving the effectiveness.

Zhang et al. proposed two strategies, namely word-level and sentence-level oracles, to select the generated results of the model [8]. However, the article continues to employ probability with decay for sampling from model's results (oracles) and the ground truth, rather than adopting a more sophisticated approach of sampling based on the quality of sentences or words.

Obviously, sampling words with equal probability regardless of their varying qualities is inherently inaccurate. It is highly recommended to employ distinct sampling probabilities for individual sentences and for each word within a sentence. Liu et al. proposed confidence-aware scheduled sampling [10], which sets two thresholds t_{golden} and t_{rand}, and samples the ground truth, model prediction result, a random word when the confidence is respectively in $[0, t_{golden})$, $[t_{golden}, t_{rand})$, $[t_{rand}, 1]$. Xu et al. [9] focused on dialogue generation and selected words based on the cosine similarity between the predicted word and the ground truth. If the similarity is greater than the threshold β, the predicted word is selected with a probability of α, which is a hyperparameter that increases with the number of training epochs to achieve faster convergence in the early stages of training and alleviate exposure bias.

However, these methods have some limitations. Firstly, they don't take into account the very important sentence-level information. Secondly, the method of threshold truncation also needs to be discussed. If probability selection is used, different words can be sampled with different probabilities, so that the distinction between different words is greater and the adaptability of the model is stronger.

Consequently, we propose a novel bilevel scheduled sampling model that effectively combines sentence-level information with word-level quality. To enhance sampling diversity and improve the model's adaptability, we introduce a smooth function that maps the integrated outcome of sentence-level and word-level information to a suitable range. Subsequently, we employ probabilistic sampling based on the mapped values instead of threshold truncation.

3 Bilevel Scheduled Sampling Model

3.1 Mathematical Modeling of Dialogue Generation

In dialogue generation tasks, we typically define the model's objective function by maximizing the conditional probability, which means we need to maximize the probability of generating the output text Y given the input text X. Specifically, we can use Eq. 1 to represent this objective function, where y_t is the t-th word in Y, $y_{<t}$ are the first $t-1$ words in Y, and T is the length of Y. This objective function requires us to calculate the probability of each word given the input text X and the previous words $y_{<t}$, and multiply these probabilities to obtain the probability of the output text Y.

$$P(Y|X,\theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, X, \theta) \tag{1}$$

During the training process, we typically use the cross-entropy loss function [14] as the optimization objective for the model, which measures the difference between the model's output and the ground truth. Specifically, we can use Eq. 2 to represent the cross-entropy loss function, where $L(\theta)$ denotes the cross-entropy loss function of the model, and the negative logarithm of the probability is used as the loss, such that the loss decreases as the probability increases.

$$L(\theta) = -\log P\left(Y|X,\theta\right) = \frac{1}{T}\sum_{t=1}^{T} -\log p\left(y_t|y_{<t}, X, \theta\right) \tag{2}$$

By minimizing the loss function of all samples, we can obtain the optimal model parameters θ. This can be expressed as Eq. 3, where N is the total number of training set samples.

$$\theta = \underset{\theta}{\operatorname{argmin}}\{\sum_{k=1}^{N} L_k(\theta)\} \tag{3}$$

In exposure bias problem, unlike training, during inference, the probability of each target word $p(y_t|y_{<t}, X, \theta)$ in Eq. 1 is conditioned on the previously generated words $y^{*}< t$ instead of the ground truth $y< t$, because the ground truth words are not available in actual inference. Therefore, we employ the scheduled sampling method to replace the ground truth during training with sampled sentences, thereby reducing the gap between the model's training and inference processes.

3.2 Bilevel Scheduled Sampling

We propose a sentence- and word-level fusion sampling method to address the diversity and contextuality of open-domain dialogue systems. Taking the transformer as an example, we introduce the principle and implementation of the bilevel scheduled sampling method, as shown in Fig. 1.

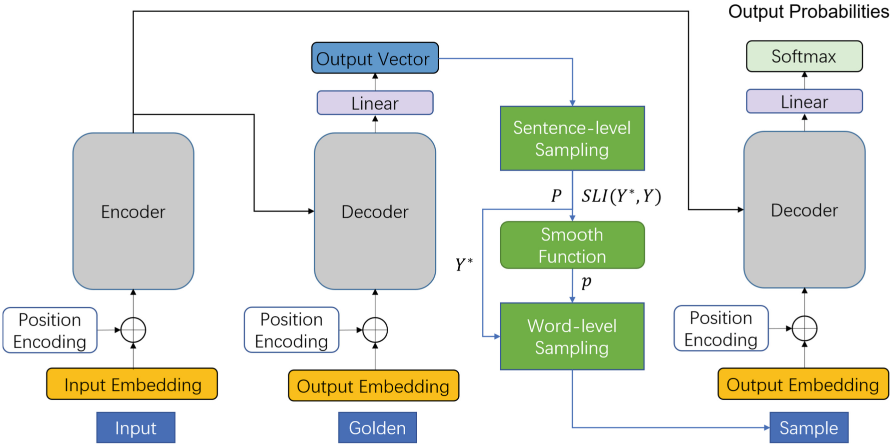


Fig. 1. Our Bilevel Scheduled Sampling model.

Sentence-Level Sampling. Zhang et al. [8] proposed a traditional sentence-level sampling method that uses beam search to select k most likely sentences from the predicted distribution, and then chooses one as the sampling sample based on the BLEU score, followed by the decay sampling method to extract from the sample and the real sentence. However, this method has two main drawbacks: (1) it requires multiple beam searches to generate sentence like inference procedure, resulting in high computational cost and low efficiency; (2) it can only sampling by probability with decay, which ignores the differences between words in the sentence, making it impossible to dynamically adjust the quality of each word.

To address these issues, we propose a sentence-level sampling method that utilizes the parallelism of the transformer model to generate the entire sentence at once during training, and calculates the probability of each word and the quality of the sentence.

We first calculate the probability P of the predicted word through the softmax function, and select the words Y^* with the largest probability to be evaluated by the sentence-level indicator.

We try two sentence-level indicators(SLI): BLEU and sentence-level cosine similarity, both of which have achieved good results. And considering the padding tokens in the sentences, we mask them during calculation. One metric based on BLEU is as follows:

$$\mathrm{SLI1}(Y^*, Y) = \frac{1}{m}\sum_{i=1}^{4} \text{bleu-i}(Y^*, Y) \tag{4}$$

where the term ‘bleu-i’ refers to the i-gram BLEU result without using a smooth function [15]. Y^* represents the model-generated result, Y represents the ground truth, and m is a hyperparameter used to map the result to a value around 1 to prevent imbalance during sampling.

The other metric based on cosine similarity is as follows:

$$\begin{aligned}\mathrm{SLI2}(Y^*, Y) &= \frac{1}{m}\,\mathrm{CosinSimilarity}(Y^*, Y) \\ &= \frac{\mathrm{embed}(Y^*)\cdot \mathrm{embed}(Y)}{m \cdot ||\,\mathrm{embed}(Y^*)|| \cdot ||\,\mathrm{embed}(Y)||}\end{aligned} \tag{5}$$

where m is the hyperparameter, which maps the result to a value around 1 so that the sampling is not unbalanced; embed is word embedding and converts word subscript to word vector. We just use word embedding within the model’s decoder. An existing word embedding from pretrained model such as BERT [16] can also be used. Here, for sentences, it is the average word embedding of sentences:

$$\mathrm{embed}(Y) = \frac{\sum_{i=1}^{T} \mathrm{embed}(y_i)}{T} \tag{6}$$

where T is the length of sentence Y, y_i is the i-th word.

This method has the following advantages: First, the whole sentence can be generated at once, avoiding the cost of multiple beam searches and improving efficiency. Second, it considers the sentence-level information and can be combined with the word-level sampling to dynamically adjust each word, enables the model to sample based on the quality of individual words while simultaneously accounting for the overall quality of the sentence.

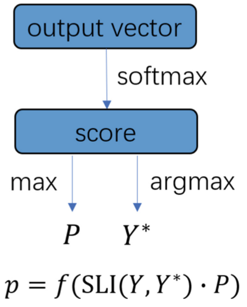


Fig. 2. Sentence-level Sampling

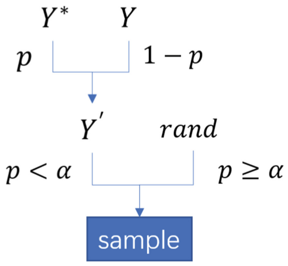


Fig. 3. Word-level Sampling

Smooth Function Between Two Sampling Layers. In order to simulate the inference process, words with higher corresponding probabilities P have a higher probability of being sampled, so we directly use P as the word-level evaluation metric. Now we have the sentence-level score S(result of SLI) and the word-level score P. However, a method to combine them is still missing. Considering that both sentence-level score S and word-level score P are equally important, we multiply them and then use a smooth function to map the product to the range of 0˜1, like Fig. 2.

Currently, we take two types of functions into account. The first one is simpler, which directly restricts the result to 0˜1, like Eq. 7.

$$f(x) = max(min(x,\, 1),\, 0) \tag{7}$$

The second one is a sigmoid-shaped function which is smoother, like Eq. 8.

$$f(x) = \frac{1}{1 + e^{-k(x-b)}} \tag{8}$$

where $k \geq 1$ is a hyperparameter to control the speed of convergence, $b > 0$ controls the central symmetrical point of the function, moving it to the right. We finally set $k = 10, b = 0.6$. Their images are in Fig. 4.

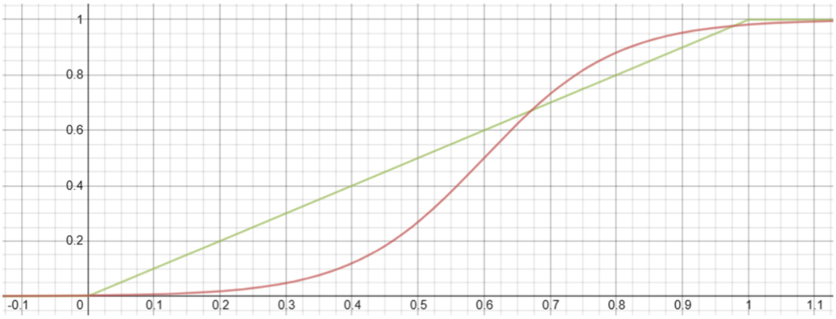


Fig. 4. The smooth function f.

Word-Level Sampling. Now, we have obtained the smoothed score p, which contains information combining sentence-level and word-level, and we use it for specific word sampling as shown in Fig. 3.

First, we use probability selection for sampling like Eq. 9, which has a greater discrimination between different words compared to threshold truncation. It can also sample different words, making the trained model more adaptable. Specifically, the predicted word Y^* is selected with probability p and the ground truth Y is selected with probability $1 - p$, and $Y^{'}$ is the first sampled result.

$$Y^{'} = \begin{cases} Y^* & sampling\ with\ probability\ p\,, \\ Y & otherwise. \end{cases} \tag{9}$$

In addition, to prevent the model from relying too heavily on high-probability predicted words and causing the generated results to be too monotonous, this paper also uses a method for random word sampling, like Eq. 10. When the word-level predicted probability P is greater than a set threshold α, a random word is chosen as the next input with a certain probability to prevent the model from degenerating.

$$sample = \begin{cases} Y' & if \;\; P < \alpha\,, \\ rand & otherwise. \end{cases} \tag{10}$$

where *rand* refers to a random word, α is a threshold which is set to 0.95, *sample* is the finally sampled result as the input to encoder during training.

4 Experiments

To assess the effectiveness and merits of the proposed Bilevel Scheduled Sampling model detailed in this study, a comprehensive set of experiments was conducted, encompassing evaluation, comparative analysis and ablation study. This section elucidates the experimental design employed and provides a thorough evaluation of the obtained results.

4.1 Datasets

We evaluate the proposed method using two widely used dialogue datasets. DailyDialog is a collection of daily life conversations, encompassing a variety of topics, emotions and linguistic styles [17]. PersonaChat consists of conversations between two participants, where one participant assumes a persona and the other participant engages in a conversation while considering the persona's characteristics [18]. After data preprocessing, we split the n-turn dialogue $(u_1, u_2, ..., u_n)$ into n-1 single-turn dialogues $[(u_1, u_2), (u_2, u_3), ..., (u_{n-1}, u_n)]$, where u represents an utterance. The number of context-response pairs in the train/validation/test set is 68,066/6,820/6,841 for DailyDialog and 104,609/12,588/12,106 for PersonaChat without any extra label or persona information.

4.2 Implementation Details

The experiment uses an NVIDIA GeForce RTX 2080 Ti graphics card and adopts PyTorch deep learning framework for training. Dropout is used for the selfattention module, the feed-forward layer, and the activation layer, and the rate of all three is set to 0.1. The sentence length is set to 26 and batch size is set to 256. The vocab size is 21626 for DialyDialog and 22630 for PersonaChat.

4.3 Comparison Methods

We compare our proposed Bilevel Scheduled Sampling model with following established methods, and all approaches are based on the Transformer-base model [11]:

- **Transformer** [11]: The Transformer-base model used in dialog generation.
- **AdapBridge** [9]: An improved scheduled sampling approach, which uses an adaptive bridge mechanism to evaluate model generation results. Specifically, the selection is made according to the cosine similarity results of the predicted word and the ground truth. If it is greater than a threshold β, the predict word is sampled according to a probability α that varies with the number of training epochs. According to the paper, we set $w = 15$ as half of the training epochs and other hyperparameters are not changed.
- **Confidence-Aware** [10]: An improved scheduled sampling method, which selects whether to sample according to the confidence of the prediction result (that is, the prediction probability). Specifically, it sets two thresholds t_{golden} and t_{rand}, and samples the ground truth, model prediction result, a random word when the confidence is respectively in $[0, t_{golden})$, $[t_{golden}, t_{rand})$, $[t_{rand}, 1]$. We set $t_{golden} = 0.7$ and $t_{rand} = 0.95$ to get the best result.

At the same time, ablation experiments are carried out in this paper. For different models, we tested different hyperparameter values for m, and presented the best result. The proposed model for ablation experiment testing is as follows:

- **Bilevel-None**: The bilevel scheduled sampling model proposed in this paper without the sentence-level sampling part. The smooth function is the sigmoid-shaped Eq. 8 and the word-level sampling method is unchanged.
- **Bilevel-Bleu**: The bilevel scheduled sampling model proposed in this paper, the sentence-level indicator is the Bleu metric with $m = 0.8$ to get the best result. The smooth function is Eq. 8.
- **Bilevel-Cosine**: The bilevel scheduled sampling model proposed in this paper, the sentence-level indicator is the sentence-level cosine similarity with $m = 0.6$ to get the best result. The smooth function is Eq. 8.
- **Bilevel-f1**: The bilevel scheduled sampling model proposed in this paper, the sentence-level indicator is the Bleu metric with $m = 0.9$ to get the best result. The smooth function is the linearly truncated Eq. 7.

4.4 Automatic Evaluation

We evaluate the performance of dialogue generation where both automatic and human evaluation metrics are applied. Automatic evaluation metrics include BLEU-1/2/3/4 [15], Distinct-1/2/3 [19]. The result is shown in Table 1.

The experimental findings demonstrate a notable performance improvement of the proposed bilevel scheduled sampling model over both the traditional scheduled sampling model and the proposed model lacking sentence-level sampling, as evidenced by the higher BLEU-1/2/3/4 scores achieved. This superiority can be attributed to the incorporation of sentence-level information in the proposed model, which leads to the generation of more coherent and natural sentences, thereby achieving better alignment with the ground truth. Furthermore, it was observed that utilizing BLEU as the sentence-level score yielded better results compared to sentence-level cosine similarity. This discrepancy may arise from

Table 1. Evaluation results on Daily Dialog and Persona Chat datasets.

Daily Dialog							
Model	BLEU-1/2/3/4				Distinct-1/2/3		
Transformer [11]	16.47	5.96	3.30	2.11	0.90	4.53	11.34
AdapBridge [9]	16.78	6.06	3.51	2.18	0.85	4.24	10.13
Confidence-Aware [10]	16.63	6.45	3.65	2.25	0.89	4.40	10.73
Bilevel-None	16.84	6.53	3.66	2.33	**0.99**	4.86	11.69
Bilevel-Bleu	**17.43**	**6.81**	**3.87**	**2.49**	0.98	**5.10**	**12.98**
Bilevel-Cosine	17.24	6.73	3.79	2.45	0.94	4.65	11.43
Persona Chat							
Model	BLEU-1/2/3/4				Distinct-1/2/3		
Transformer [11]	17.79	6.37	3.42	2.31	0.22	0.65	1.34
AdapBridge [9]	19.53	6.79	3.65	2.46	0.20	0.58	1.20
Confidence-Aware [10]	20.15	7.35	3.82	2.48	0.19	0.62	1.36
Bilevel-None	19.84	7.20	3.79	2.48	0.20	0.67	1.47
Bilevel-Bleu	21.16	**7.79**	**4.10**	**2.71**	**0.22**	**0.70**	**1.61**
Bilevel-Cosine	**21.17**	7.74	4.10	2.68	0.21	0.64	1.34

the fact that BLEU places greater emphasis on text matching, while cosine similarity focuses more on semantic alignment, thus favoring the former for improved BLEU results.

Additionally, the proposed model shows an advantage in the Distinct-1/2/3 metric. This advantage stems from the incorporation of sentence-level sampling information and the utilization of probabilistic sampling for word generation. The bilevel model effectively generates sentences that are more diverse, mitigating the issue of excessively repetitive or singular output, and consequently attains higher scores in the Distinct metric.

In summary, the experimental results demonstrate that by combining sentence-level and word-level considered together with probabilistic sampling, the proposed model outperforms existing scheduled sampling models in terms of both BLEU-1/2/3/4 and Distinct-1/2/3 metrics. This indicates that the bilevel scheduled sampling model significantly alleviates the exposure bias problem and outperforms the state-of-the-art scheduled sampling methods.

4.5 Human Evaluation

To thoroughly assess the proposed model and the baseline model mentioned in this paper, we conducted a human evaluation following the approach used by Li et al. [20]. For this evaluation, we randomly selected 100 samples from the test set of each dialogue dataset. Subsequently, we sought the judgment of three well-educated annotators to determine whether the overall response quality of the Bilevel-Bleu model and the other models under consideration exhibited superior coherence, informativeness, and fluency. The annotators categorized their assessment as either a win, tie, or lose for each model.

Table 2. Human evaluation result.

Bilevel-Bleu vs. Models	Win	Tie	Lose	Kappa
Transformer [11]	60.33	25.50	14.17	0.6581
AdapBridge [9]	48.50	29.67	21.83	0.5417
Confidence-Aware [10]	43.00	32.17	24.83	0.5077
Bilevel-None	44.50	32.00	23.50	0.4491
Bilevel-Cosine	39.33	33.33	27.33	0.5223

Table 2 summarizes the human evaluation results. The final results show that the Bilevel-Bleu model in this paper is better than other models, which indicates that it is more capable of generating human preferred responses. Meanwhile, we use Fleiss kappa [21] to measure the agreement between annotators, and the results are all greater than 0.4, which indicates that the annotators reach a good agreement on the judgment.

4.6 Ablation Study

In this paper, we design a comparison experiment includes the proposed model without sentence-level sampling and with two different sentence-level sampling. As can be seen from Table 1, whether the sentence-level indicator uses bleu or cosine similarity, the model combining sentence-level and word-level sampling in this paper will have better results than the single word-level sampling model. At the same time, it can also be seen from the results that the performance of the proposed model is significantly improved compared to the base transformer model when only word-level sampling is performed, with slightly better results than the existing scheduled sampling methods. This indicates that the probabilistic sampling approach in this paper is better than threshold truncation if we map the original probability to a suitable size by smoothing function.

In addition, we conducted comparative experiments on smooth functions, including linear truncation function f1 (Eq. 7) and sigmoid-shaped smooth function f2 (Eq. 8). The result is in Table 3.

The results indicate that the sigmoid-shaped smooth function gives better results, enabling the model to sample more appropriate sentences. This is due to the direct probabilistic sampling is not sufficient to distinguish the high-quality and low-quality utterances. By employing the sigmoid-shaped smooth function, the influence of sentence performance on the sampling probability is enhanced, resulting in a smoother and more effective sampling process.

Table 3. Evaluation results of different smooth functions on the PersonaChat dataset.

Model	BLEU-1/2/3/4				Distinct-1/2/3		
Transformer [11]	17.79	6.37	3.42	2.31	0.22	0.65	1.34
Bilevel-f1	19.58	7.18	3.79	2.45	0.19	0.64	1.35
Bilevel-f2	**21.16**	**7.79**	**4.10**	**2.71**	**0.22**	**0.70**	**1.61**

5 Conclusion

In this paper, we propose a bilevel scheduled sampling model, which considers the sentence-level and word-level combination quality of the model generation results, so that the sampling results can better adapt to the exposure bias and thus improve the performance of the model. In order to make the sampling more diverse and improve the adaptability of model, we propose a smooth function to map the combined result of sentence-level and word-level to an appropriate range, and then perform probabilistic sampling instead of threshold truncation. Experiments on two widely used open-domain dialogue datasets demonstrate the effectiveness of all our proposed methods, which significantly alleviate the exposure bias problem and outperform state-of-the-art scheduled sampling methods. In the future, we plan to extend the application of the Bilevel Scheduled Sampling method to large language models across various projects, addressing the issue of exposure bias. This approach will help enhance the performance and robustness of the models in real-world scenarios.

Acknowledgement. This research was supported by the Beijing Natural Science Foundation (No. 4222037, L181010).

References

1. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
2. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. arXiv preprint arXiv:1606.02960 (2016)
3. Venkatraman, A., Hebert, M.H., Bagnell, J.A.: Improving multi-step prediction of learned time series models. In: National Conference on Artificial Intelligence. AAAI Press (2015)
4. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. Comput. Sci. (2015)
5. Mihaylova, T., Martins, A.F.: Scheduled sampling for transformers. arXiv preprint arXiv:1906.07651 (2019)
6. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence (2017)

7. Nie, W., Narodytska, N., Patel, A.: RelGAN: relational generative adversarial networks for text generation. In: International Conference on Learning Representations (2018)
8. Zhang, W., Feng, Y., Meng, F., You, D., Liu, Q.: Bridging the gap between training and inference for neural machine translation. arXiv preprint arXiv:1906.02448 (2019)
9. Xu, H., Zhang, H., Zou, Y., Chen, H., Ding, Z., Lan, Y.: Adaptive bridge between training and inference for dialogue generation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 2541–2550 (2021)
10. Liu, Y., Meng, F., Chen, Y., Xu, J., Zhou, J.: Confidence-aware scheduled sampling for neural machine translation. arXiv preprint arXiv:2107.10427 (2021)
11. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
12. Goodman, S., Ding, N., Soricut, R.: TeaForN: teacher-forcing with N-grams. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8704–8717 (2020)
13. Liu, Y., Meng, F., Chen, Y., Xu, J., Zhou, J.: Scheduled sampling based on decoding steps for neural machine translation. arXiv preprint arXiv:2108.12963 (2021)
14. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cogn. Sci. **9**(1), 147–169 (1985)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
17. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: a manually labelled multi-turn dialogue dataset. arXiv preprint arXiv:1710.03957 (2017)
18. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? arXiv preprint arXiv:1801.07243 (2018)
19. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055 (2015)
20. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547 (2017)
21. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378 (1971)

Dial-QP: A Multi-tasking and Keyword-Guided Approach for Enhancing Conversational Query Production

Jiong Yu[1,2], Sixing Wu[1,2](✉), Shuoxin Wang[1,2], Haosen Lai[1,2], and Wei Zhou[1,2]

[1] Engineering Research Center of Cyberspace, Yunnan University, Kunming, China
yujiong@mail.ynu.edu.cn,
{wusixing,zwei}@ynu.edu.cn
[2] National Pilot School of Software, Yunnan University, Kunming, China

Abstract. Recent works have utilized a dynamic process to seek up-to-date knowledge from search engines, yielding promising performance improvement in the field of knowledge-grounded response generation models. This pipeline consists of two stages: a *Conversational Query Production (CQP)* stage to acquire knowledge and a *Response Generation* to generate the final response. In this paper, we focus on the CQP task, which aims to condense the dialogue context into a concise query that is then fed to search engines. Previous studies have treated the problem as a generative task and always generated queries solely based on the dialogue context in an end-to-end manner. However, such a straightforward approach suffers from the followings: 1) the CQP model is hard to determine whether generating a query is necessary since not all scenarios require retrieval knowledge; 2) redundant content in the dialogue context can make it difficult for the model to recognize the key information that reflects the user's concerns, particularly in multi-turn conversations. To address these challenges, we propose a novel BART-based *Dial-QP* for the CQP task, which decomposes the CQP task into *Query Classification* and *Query Production* tasks and then introduces a multitasking training paradigm to improve the decision-making process of *No Query*. Meanwhile, a *Keyword Recognition* stage helps the model focus on the key information. Extensive experiments and analyses on two CQP datasets DuSinc and WoI have demonstrated the effectiveness of *Dial-QP*. Our code is available at: https://github.com/Y-NLP/Chatbots/tree/main/NLPCC2023_Dial-QP.

Keywords: Conversational query production · Knowledge-grounded response generation

1 Introduction

Knowledge-grounded response generation (KRG) models can generate context-appropriate and informative dialogue responses with the help of extra knowledge besides the dialogue context [20,21]. However, KRG faces challenges in

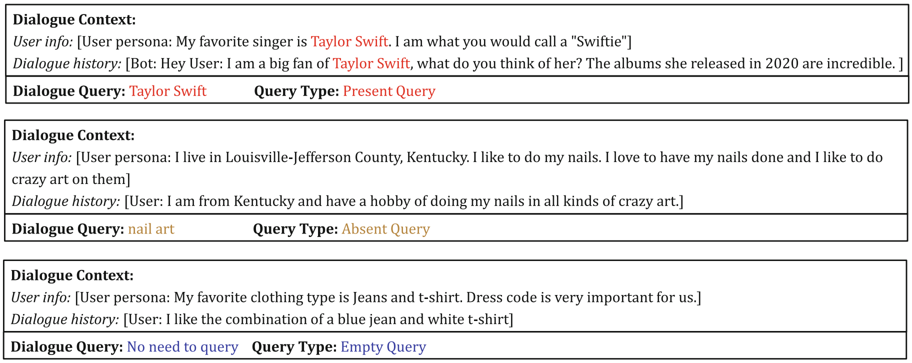
F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 840–852, 2023.
https://doi.org/10.1007/978-3-031-44693-1_65

accessing up-to-date information and achieving sufficient knowledge coverage since the majority of existing works are only able to retrieve information from static knowledge bases. To break this bottleneck, recent works have attempted to dynamically seek up-to-date knowledge from search engines [8,16,24]. Search engines such as Google and Bing have scanned and indexed billions of web pages and thus can provide a vast amount of the newest information on a wide range of topics. Nevertheless, this abundance of knowledge in search engines also poses significant challenges for downstream tasks to retrieve appropriate information. Consequently, a two-stage pipeline is often used for such KRG models to retrieve knowledge from search engines efficiently and accurately. The first stage, *Conversational Query Production*, learns to retrieve knowledge from search engines by generating queries, while the next stage, *Response Generation*, utilizes the retrieved knowledge to generate the response. Moreover, practical applications like the recent New Bing[1], a generative chat-style search engine integrated with large language models, have already shown powerful capabilities, further demonstrating the effectiveness of this search-then-generate pipeline in real-world scenarios.

This work focuses on the *Conversational Query Production (CQP)* stage of KRG models. Previous works, including BlenderBot2 [18], PLATO-SINC [1] and SeeKeR [16] have already yielded promising results in the KRG domain by retrieving real-time knowledge from search engines using CQP. However, these methods simply reply on backbone PLMs like the auto-regressive GPT-2 [16] or the sequence-to-sequence BART [8] to generate queries solely based on the dialogue context, which poses two challenges in the practical scenarios. Firstly, some scenarios do not require knowledge retrieval from the Internet, such as the third case in Fig. 1. Therefore, CQP models should have the capability to determine whether generating a query is necessary. Prior works [24] have instructed the model to generate a special symbol, *No Query*, for decision-making. This solution is simple but can cause models to over-predict the *No Query* symbol because massive *No Query* training instances may bring biases in the end-to-end *Maximum Likelihood Estimation* learning [7]. Secondly, in multi-turn conversations, the dialogue context can accumulate redundant content, making it challenging for the model to discern the key information that the user is really focused on. Thus, it becomes difficult for the model to generate relevant queries that address the user's primary concerns. These challenges may impact the quality of the generated queries and the subsequent *Knowledge-Grounded Response Generation* task.

Given the substantial impact of these challenges, we propose a novel BART-based [9] approach for conversational query production, called *Dial-QP*. To balance the query generation and decision-making process of *No Query*, Dial-QP divides the CQP task into *Query Classification* and *Query Production* tasks. Specifically, the *Query Classification* task learns to estimate the target query type in the backbone encoder, while the *Query Production* task generates non-empty queries in a generative manner. We introduce a multitasking training

[1] https://www.bing.com/new.



Dialogue Context:
User info: [User persona: My favorite singer is Taylor Swift. I am what you would call a "Swiftie"]
Dialogue history: [Bot: Hey User: I am a big fan of Taylor Swift, what do you think of her? The albums she released in 2020 are incredible.]

Dialogue Query: Taylor Swift **Query Type:** Present Query

Dialogue Context:
User info: [User persona: I live in Louisville-Jefferson County, Kentucky. I like to do my nails. I love to have my nails done and I like to do crazy art on them]
Dialogue history: [User: I am from Kentucky and have a hobby of doing my nails in all kinds of crazy art.]

Dialogue Query: nail art **Query Type:** Absent Query

Dialogue Context:
User info: [User persona: My favorite clothing type is Jeans and t-shirt. Dress code is very important for us.]
Dialogue history: [User: I like the combination of a blue jean and white t-shirt]

Dialogue Query: No need to query **Query Type:** Empty Query

Fig. 1. Queries can be classified as three types in CQP: Present Query is a continuous substring of the dialogue context, Absent Query can not be directly given from the dialogue context, and Empty Query means do not need a query. (Color figure online)

paradigm for the joint learning of both tasks by sharing the same backbone BART. By employing this multitasking training paradigm, *Dial-QP* can accurately determine when to generate a *No Query* based on the context and generate an appropriate query only when needed. Additionally, to address the impact of redundant content in multi-turn conversations, we propose the incorporation of contextual hinting keywords (i.e., *Keyword Recognition*) to help the model focus better on the key information in the dialogue context. Building on this concept, this stage helps the model generate more relevant and targeted queries.

We conduct extensive experiments on two CQP datasets: a Chinese *DuSinc* [24] and an English *WoI* [8]. Experimental results prove *Dial-QP* can significantly outperform baselines in the *Query Classification* and *Query Production* tasks. Additionally, we provide extensive analyses to investigate our approach further. The contribution of this work is threefold: 1) We propose to classify the query into different types, which is a more fine-grained and reasonable perspective to process the CQP task; 2) We introduce a novel approach, *Dial-QP*, to conduct the CQP task, which incorporates a multitasking training paradigm for improving the decision-making process of *No Query* and a *Keyword Recognition* stage to help the model focus on the potential key information. Together, these techniques contribute to the generation of correct and relevant queries. 3) Our extensive experiments and analyses demonstrate the performance of *Dial-QP*.

2 Related Work

Conversational Query Production (CQP) is a precursor task for the web knowledge-grounded dialogue generation works. Its goal is to condense dialogue context into concise phrases, which are then used as a query to search for knowledge from the internet. For instance, given a context such as ‘User: Who is your favourite singer? Bot: Taylor Swift. User: What song of hers do you like to listen to?’, the output ‘Taylor Swift songs’ can be used as a query to search for knowledge. In detail, the CQP task was first introduced by [8]. It involved an internet-augmented dialogue generation paradigm. Following this work, BlenderBot2 [18], PLATO-SINC [24], and SeeKeR [16] have been proposed and demonstrated the potential of CQP in the subsequent knowledge-grounded dialogue response generation. However, most of the present works [8,16,24] soely conduct the CQP task in an end-to-end manner, which employ the backbone PLMs (e.g., BART [9], GPT-2 [14], T5 [15], and PLATO-XL [1]) to directly generate queries. Furthermore, DuSinc [24] and WoI [8] are two common CQP datasets in Chinese and English respectively. These datasets are collected from human-human conversations in which one participant decides whether to generate a query to retrieve knowledge.

Keyphrase Prediction (KP) is a task closely related to CQP, as it aims to obtain a set of Keyphrases that highlight the core topics, or information of a given document [19]. Therefore, many KP methods can also be used in the field of CQP. KP methods can be divided into two categories: Keyphrase Extraction (KE) and Keyphrase Generation (KG). Generally, KE can be conducted in an unsupervised manner and is divided into three main types: statistical-based [2,5], graph-based [6,12], and embedding-based [17]. However, KE methods can not judge whether an empty *No Query* is needed. On the other hand, KG often employs a sequence-to-sequence generation model, and the backbone model has evolved from RNN [3,11] to transformer [22,23] to PLMs [4,19]. Although they can predict *No Query* in the generation, they also face challenges in the over-generated *No Query* issue.

3 Methodology

3.1 Problem Definition

Given a corpus $\mathcal{D} = \{(C, Q, Y)\}^N$ of N instances, where $C = \{c_1, c_2, \cdots, c_n\}$ is the dialogue context consisting of the background user information and dialogue histories, $Q = \{q_1, q_2, \cdots, q_m\}$ is the target dialogue query, and Y is the label of the corresponding query type. Thus, the conversational query production task is to generate a dialogue query Q based on the dialogue context C, namely maximizes the probability: $\prod_{t=1}^{m} P_{CQP}\left(q_t \mid q_{<t}, C\right)$.

According to the features of dialogue query, the target query Q can be classified into three kinds (i.e., Y) in Fig. 1:

- *Present Query* Q^P *:* It is a continuous substring of the dialogue context C;
- *Absent Query* Q^A*:* It does not match any contiguous substring of C, but may have some partial overlap with C.
- *Empty Query* Q^E*:* It is a special symbol to indicate *No Query* is needed. For example, for a simple dialogue history 'Hello!'.

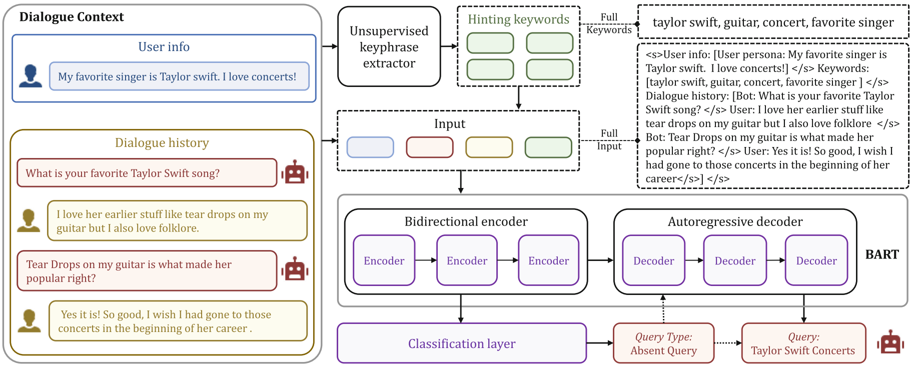

Dialogue Context
User info
My favorite singer is Taylor swift. I love concerts!
Dialogue history
What is your favorite Taylor Swift song?
I love her earlier stuff like tear drops on my guitar but I also love folklore.
Tear Drops on my guitar is what made her popular right?
Yes it is! So good, I wish I had gone to those concerts in the beginning of her career .
Unsupervised keyphrase extractor
Hinting keywords
Full Keywords
taylor swift, guitar, concert, favorite singer
Input
Full Input
<s>User info: [User persona: My favorite singer is Taylor swift. I love concerts!] </s> Keywords: [taylor swift, guitar, concert, favorite singer] </s> Dialogue history: [Bot: What is your favorite Taylor Swift song? </s> User: I love her earlier stuff like tear drops on my guitar but I also love folklore </s> Bot: Tear Drops on my guitar is what made her popular right? </s> User: Yes it is! So good, I wish I had gone to those concerts in the beginning of her career</s>] </s>
Bidirectional encoder
Encoder
Encoder
Encoder
Autoregressive decoder
Decoder
Decoder
Decoder
BART
Classification layer
Query Type:
Absent Query
Query:
Taylor Swift Concerts

Fig. 2. The illustration of *Dial-QP*.

3.2 Dial-QP

As illustrated in Fig. 2, this work proposes a BART-based *Dial-QP* to generate the query Q accurately. The frequently appeared empty query training instances make the backbone model tend to generate empty queries in the inference stage as well. To address this issue, we designed a multitasking training paradigm for joint learning of *Query Classification* and *Query Production*, instead of generating an empty query as previous works did. Besides, to guide the model focus on the key information within the C, we involve a *Keyword Recognition* stage to enrich the context by identifying potential keywords as contextual hints.

Keyword Recognition. As mentioned before, key information is necessary to be identified in the dialogue context to reflect the user's concerns. However, redundant content in the dialogue context can make it challenging to identify this key information, especially in multi-turn conversations. By incorporating additional guidance content, it is possible to mitigate this negative impact. Thus, we identify some contextual hinting keywords to guide *Dial-QP*.

We use SIFRank [17], a powerful unsupervised Keyphrase extraction method based on the PLMs, to identify a list of contextual hinting keywords $S = \{s_1, s_2, \cdots, s_l\}$ from the dialogue context C. According to our statistics, there is a high overlap between the golden query and the dialogue context. In fact, even for absent queries that do not entirely originate from the context, the word-level overlap between the query and the context reaches 71% and 65% in DuSinc and WoI. Therefore, hinting keywords not only helps the model focus on potential key information but also contributes to forming the final query more accurately.

Query Classification. Not all conversational scenarios need to use knowledge from the Internet, such as simple greetings. Then, a large number of empty queries Q^E will inevitably appear. For instance, about 41% and 15% of scenarios do not need a query in the DuSinc and WoI, respectively. To handle this issue, previous works [24] often ask the backbone model to directly generate a special symbol *No Query* to indicate such empty queries. However, similar to the response generation models that tend to generate generic responses due to the massive training instances are generic [10], this paradigm will lead the backbone model tends to generate *No Query* no matter whether an informative query is really needed or not.

Thus, rather than directly generating *No Query*, we introduce a *Query Classification* task to predict the query type. Formally, we adopt the BART encoder (denoted as $BART_{Enc}$) to estimate the target query type:

$$\hat{Y} = \theta\left(\mathbf{W_R}\left(BART_{Enc}\left([CLS], \omega_C, [SEP], \omega_S\right)\right)\right) \tag{1}$$

where $\hat{Y}$ is the predicted label, θ is the $softmax$ function, $BART_{Enc}$ outputs the last hidden state of $[CLS]$; $\mathbf{W_R}$ is a learn-able linear classifier; ω_C and ω_S linearize C and S into formatted plain texts.

Query Production. As a generative task, we use the same BART to generate the dialogue query Q based on the dialogue context C, the recognized keyword set S, and the query type Y. Specifically, we feed the output of $BART_{Enc}$ to the BART decoder (denoted as $BART_{Dec}$), then estimate $P_{QP}\left(q_t \mid q_{<t}, C, S\right)$ and finally generate the Q:

$$Q = BART_{Dec}(BART_{Enc}([CLS], \omega_C, \omega_S, [SEP])) \tag{2}$$

Multi-task Learning. In the training stage, we employ a multitasking training paradigm for the joint learning of the above two tasks. Specifically, we first target the *Query Classification* task, forcing the query classifier learning to consider whether the dialogue query is *No Query* by optimizing the following cross-entropy loss function with the golden label Y:

$$\mathcal{L}_{\mathcal{QC}} = -\frac{1}{N}\sum_{i=1}^{N} Y_i \log(\hat{Y}_i) \tag{3}$$

where $\hat{Y}$ is the predicted label.

If the golden label of the given instance is a present query Q^P or an absent query Q^n, then this instance will enter the next training process of *Query Production* task. Subsequently, the following objective function tries to generate the dialogue query based on all the information:

$$\mathcal{L}_{\mathcal{QG}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{q_{i,t}\in Q_i} \log P_{QP}\left(q_{i,t} \mid q_{i,<t}, C_i, S_i, \hat{Y}_i\right) \tag{4}$$

Then, the global multitasking training loss is formulated as follows:

$$\mathcal{L}_{\mathcal{CQP}} = \lambda_C \cdot \mathcal{L}_{\mathcal{QC}} + \lambda_G \cdot \mathcal{L}_{\mathcal{QG}} \tag{5}$$

where λ_C and λ_G are the weights for the two tasks. In practice, we set λ_C and λ_G as 5.0 and 1.0 respectively to balance the loss gap between the two tasks.

In the inference stage, our *Dial-QP* model can accurately estimate the target query type and produces suitable queries for various query types. Additionally, since the query production module is trained on non-empty query data, we can also input an empty query instance to generate a distantly supervised query, which may serve as a candidate for a dialogue query in some cases.

4 Experiment

4.1 Settings

Dataset. We conducted extensive experiments on two representative query production datasets: the Chinese *DuSinc* [24] and the English *WoI* [8]. In Chinese *DuSinc*, the training/valid/test set has 16382/500/500 instances, where the present/absent/empty query proportion is 19.29%/39.65%/41.06%, and the query average char-level length is 4.06; For the English *WoI*, its training/valid/test set has 41476/2881/2466 instances, the present/absent/empty query proportion is 30.41%/54.84%/14.75%, and the query average char-level length is 16.56. Note that both datasets are reorganized versions by a data processing stage. We regard the last dialogue query in each turn as the golden query and extract all context-query pairs to construct the training data.

Models. We select three kinds of methods as baselines:

1) Keyphrase Extraction: We select a naive statistical method *TF-IDF* [5] and an improved *YAKE* [2]; a graph-based *TextRank* [12]; an embedding-based *SIFRank* [17]. Here, we take the first-rank keyword as the predicted query.

2) Keyphrase Generation: Several SOTA methods are compared: *CatSeq* [11], *CatSeqTG* [3], and *Wr-SetTrans* [22]. We use the same process to extract keywords from the inference results same as the above methods.

3) CQP-PLMs: Most prior CQP works use Seq2Seq models; thus, we make a comparison with several representative PLMs using the HuggingFace and the PyTorch. For the Chinese *DuSinc*, we select BART *fnlp/bart-base-chinese* and EVA2.0 *thu-coai/EVA2.0-base*. For the English *WoI*, we use BART *facebook/bart-base*, and the SOTA *BlenderBot2* [18] offered by ParlAI [13].

4) Ours: For our *Dial-QP*, the backbone PLM is also based on the English BART *facebook/bart-base* or the Chinese *fnlp/bart-base-chinese*. We changed the learning rate to 5.5e-5 for joint learning the multi-training tasks.

All experiments were run on one Nvidia A100 with 40GB V-RAM.

Metrics. We use several char-level metrics to evaluate models performance in the *Query Classification* and *Query Production* tasks respectively. For the first *Query Classification* task, we take this task as a binary classification object: whether an empty query is needed or not, and leverage the following metrics: 1) *Len*: the averaged char-level length (the length of *No Query* is regarded as 1);

2)*EQP*: the proportion of empty query; 3) *QC-P/R/F1*: the precision, recall, and micro-F1 for this classification task. For the next generative *Query Production* task, we adopt 1) the overlap-based *CharF1*, *BLEU-1/2/3/4*, *ROUGE*, and the embedding-based *EM-A/G/X* to evaluate the relevance between the generated query and the golden query; 2)*DIST-1/2* to evaluate the diversity of generated query; and 3) the geometric *Mean* to measure overall performance.

Table 1. Query Classification Evaluation. *Extraction-based* methods extract queries without predicting the empty query and have the same P/R/F1; thus, we only report one result here. *Reference* is the ground truth. **First**/Second is the first/second best.

Datasets	*DuSinc*					*WoI*				
Model	*Len*	*EQP(%)*	*QC-P*	*QC-R*	*QC-F1*	*Len*	*EQP(%)*	*QC-P*	*QC-R*	*QC-F1*
Reference	4.06	41.06	100	100	100	16.56	14.75	100	100	100
SIFRank	3.77	0.0	60.6	**100**	75.5	15.63	0.0	87.5	**100**	**93.3**
CatSeq	2.68	61.0	76.9	49.5	60.2	7.73	37.7	88.3	62.9	73.5
CatSeqTG	3.78	65.0	75.4	43.6	55.2	7.73	37.7	**88.4**	62.9	73.5
Wr-SetTrans	1.66	78.8	77.6	27.1	40.1	14.97	0.1	87.5	99.9	**93.3**
BART-Base	3.14	52.8	75.4	58.8	66.0	6.97	56.7	88.2	43.8	58.5
EVA2.0-Base	4.24	64.8	66.3	38.6	48.8	-	-	-	-	-
BlenderBot2	-	-	-	-	-	14.33	0.0	87.5	**100.0**	**93.3**
Dial-QP	**4.18**	**38.0**	**78.7**	80.5	**79.6**	**15.69**	**1.6**	87.5	98.4	92.7

4.2 Result and Analysis

Query Classification. Table 1 reports the results on the *Query Classification* task. In the *DuSinc*, most non-extraction baselines are struggled in dealing with *No Query*: 1) Keyphrase generation models and CQP-PLMs tend to over-predict the *No Query*, and thus the corresponding *EQP* is much higher than the real proportion, demonstrating such models are easy to generate this generic query; 2) Keyphrase extraction models can not generate any *No Query*. Thus, they have excellent recall but poor precision. Compared to baselines, *Dial-QP* has achieved the first or the second best in all metrics, demonstrating *Dial-QP* can predict queries more in line with the golden queries distribution. In the English *WoI*, the proportion of *No Query* is significantly lower than in the *DuSinc*. Thus, the performance differences among models are not as significant as in the previous *DuSinc*. Nonetheless, Our *Dial-QP* still has the tier-1 overall performance.

Query Production. Table 2 reports the experimental results on the Chinese *DuSinc*. It can be seen that *Dial-QP* can significantly beat the whole baselines in most metrics, demonstrating the effectiveness of our keyword recognition strategy and multi-task learning paradigm. We can find that the Keyphrase generation methods have better performance than the keyphrase-extraction methods,

Table 2. Query Production Evaluation on Chinese *DuSinc* dataset. **First**/Second also represents the first/second best performance.

Model	*CharF1*	*BLEU-1/2/3/4*	*ROUGE*	*EM-A/G/X*	*DI-1/2*	*Mean*
TF-IDF	11.2	19.7/14.4/10.6/8.3	26.3	0.64/0.64/0.55	22.74/42.2	14.3
YAKE	5.8	18.7/8.1/5.5/4.3	18.6	0.64/0.60/0.54	11.08/25.7	9.0
TextRank	12.8	26.0/15.1/11.6/9.6	26.1	0.67/0.65/0.54	26.12/59.4	17.1
SIFRank	17.2	33.0/20.0/15.2/12.7	31.4	0.65/0.67/0.58	30.65/67.4	20.9
CatSeq	35.3	40.9/32.7/27.4/23.9	55.5	0.79/0.80/0.74	29.93/60.9	28.1
CatSeqTG	27.3	40.9/30.0/22.2/17.1	53.2	0.77/0.79/0.72	21.92/42.7	23.4
Wr-SetTrans	24.3	25.9/18.6/12.2/8.9	52.0	0.76/0.79/0.71	20.42/44.2	19.0
BART-Base	44.7	50.5/42.5/37.4/33.4	58.6	0.81/0.82/0.76	**31.87**/65.8	33.4
EVA2.0-Base	11.3	28.1/15.7/6.5/3.4	41.3	0.72/0.73/0.66	23.40/57.1	17.2
Dial-QP	**49.5**	**58.7/49.5/42.8/38.0**	**61.4**	**0.84/0.83/0.79**	29.95/**69.0**	**36.5**

Table 3. Query Production Evaluation on English *WoI* dataset.

Model	*CharF1*	*BLEU-1/2/3/4*	*ROUGE*	*EM-A/G/X*	*DI-1/2*	*Mean*
TF-IDF	37.1	21.5/15.0/12.5/11.0	31.0	0.786/0.80/0.668	0.19/2.6	12.1
YAKE	43.7	41.7/22.7/15.8/12.5	31.8	0.806/0.80/0.677	0.10/1.4	15.6
TextRank	45.7	48.9/29.4/21.7/17.6	33.3	0.810/0.81/0.694	0.10/1.7	18.2
SIFRank	47.8	50.4/34.3/27.4/23.6	36.8	**0.814**/0.82/0.703	0.11/1.7	20.4
CatSeq	32.4	21.9/16.0/13.6/12.0	30.6	0.558/0.74/0.491	0.29/4.7	12.1
CatSeqTG	32.4	21.9/16.0/13.5/12.0	30.6	0.558/0.74/0.491	0.30/4.7	12.1
Wr-SetTrans	41.2	42.8/27.7/21.4/17.6	33.5	0.797/0.81/0.684	0.18/3.0	17.2
BART-Base	32.6	19.3/15.4/13.9/12.8	31.4	0.438/0.72/0.400	**0.38/5.4**	12.1
BlenderBot2	51.2	51.2/39.3/33.8/30.4	41.4	**0.814**/0.84/0.716	0.21/3.6	23.0
Dial-QP	**53.6**	**55.5/44.4/39.0/35.4**	**44.5**	0.810/**0.85/0.722**	0.19/3.3	**25.3**

indicating the query production task is more suitable for using generation-based methods to perform, rather than only capturing the superficial features in the context. EVA2.0 have the worst performance in the PLMs methods because they tend to generate too many *No Query*. The BART model achieves a notable result among baselines, showing the ability of the pre-trained language model.

As reported in Table 3, *Dial-QP* still achieves the best overall performance in the English *WoI* dataset, and exceeds the powerful SOTA BlenderBot2 in most metrics, showing the dialogue query generated by *Dial-QP* are more semantically relevant to the ground truth. *Dial-QP* is still behind some models in terms of diversity, which we owe to the hinting keywords bringing certain restrictions.

Ablation Study. We have conducted an ablation study to further analyze the effect of keyword recognition strategy and multi-task learning paradigm. We tested variants ablated model of *Dial-QP* on the Chinese *DuSinc* dataset:

1. In *w/o 3-class classifier*, we remove the 3-class classifier and the *Query Classification* task to only use the hinting keywords to generate the dialogue queries. Notably, there is a significant performance regression, especially in the *Query*

Table 4. Ablation Studies. Except for ↑, the results of variants have decreased.

Setting	*EQP(%)*	*QC-F1*	*CharF1*	*BLEU-4*	*ROUGE*	*EM-X*	*DI-2*
Full	38.0	79.6	49.50	38.0	61.4	0.79	69.0
w/o 3-class classifier	58.2	64.1	45.09	32.8	59.9	0.77	66.9
w/o Keywords	43.6	76.6	47.50	34.8	61.6↑	0.78	70.2↑
w/o All (BART)	52.8	66.1	45.01	33.4	58.6	0.76	65.8
BART with 2-class classifier	41.0	76.3	46.81	35.4	59.8	0.78	69.7↑

Classification task metrics. It means our *Query Classification* task can teach the query production model to effectively teach the query production model to predict the query type;

2. In *w/o Keywords*, we train the backbone model using multi-task learning without the help of hinting keywords. The performance overall decreased but has slightly improved in both *ROUGE* and *DI-2*, demonstrating that at the same time as the hinting keywords guide the query generation process, it may also lead to limiting the generation of various queries;
3. In *w/o All (BART)*, we discard all further processing methods, and the performance is worse than previous model variants, showing the necessity to joint use our keyword recognition strategy and multi-task learning paradigm;
4. We further test the backbone BART with a binary classifier, which only learns to distinguish whether a query is needed for one context. Compared with the *w/o Keywords*, we can find a certain performance decrease, illustrating that a fine-grained classification object is more suitable for this task.

Case Study. We report three real cases in Table 5. In cases 1 and 3, we can find *Dial-QP* has recognized the hinting keywords and successfully classified the query type; meanwhile, it also generated correct dialogue queries. Such two cases verify the effectiveness of our *Keyword Recognition*, *Query Classification*, and the final *Query Production* at the same time. In the second case, *Dial-QP* generated a contextually appropriate query by making certain inferences rather than directly extracting words from the context, although it had made a mistake in the query type. It shows our paradigm has mistake redundancy to the recognized keywords and the results predicted by the classification stage to some extent. Unlike our approach, baselines have worse performances. Keyphrase extraction methods can not well handle empty queries, while other generative models tend to over-predict the *No Query*.

Table 5. Case Study on the Chinese *DuSinc* and English *WoI* dataset. For brevity, we only present the results for some representative methods.

Case #1	用户信息*:* [话题: 泛娱乐,电影, 位置: 河南省安阳市] 对话历史*:* [用户: 你听说过《你好，李焕英》吗? ... 机器人: 当然听说过呀，这可是贾玲作为导演的处女作呢 用户: 你觉得这部电影怎么样?] *User info:* [Topic: General Entertainment, Movies, Location: Anyang, Henan] *Dialogue history:* [User: Have you heard of "Hi, Mom"? ... Bot: Of course, I've heard of it! It's Ling Jia's directorial debut. User: What do you think of this movie?]		
Golden	*Query Type:* Present Query; *Query*: 《你好，李焕英》 *Query Type:* Present Query; *Query*: "Hi, Mom"		
TextRank	河南省安阳市 Anyang, Henan	SIFRank	电影 Movies
CatSeqTG	李焕英 Huanying Li	Wr-SetTrans	贾玲电影 Ling Jia's movie
BART	贾玲电影 Ling Jia's movie	EVA2.0	不需要查询 No need to query
Dial-QP	*Keywords:* 电影, 李焕英; *Predicted Query Type:* Present Query *Predicted Query*: 《你好，李焕英》 *Keywords:* Movies, Huanying Li; *Predicted Query Type:* Present Query *Predicted Query*: "Hi, Mom"		
Case #2	*User info:* [User persona: I live in Louisville-Jefferson County, Kentucky. I like to do my nails. I love to have my nails done and I like to do crazy art on them] *Dialogue history:* [User: I am from Kentucky and have a hobby of doing my nails in all kinds of crazy art.]		
Golden	*Query Type:* Absent Query; *Query*: nail art		
TextRank	crazy art	SIFRank	crazy art
CatSeqTG	Kentucky	Wr-SetTrans	art in art
BART	No need to query	Blenderbot2	Kentucky
Dial-QP	*Keywords:* crazy art, nail; *Predicted Query Type:* Present Query *Predicted Query*: nail art		
Case #3	用户信息*:* [话题: 购物,水果,芒果 位置: 山东省泰安市] 对话历史*:* [用户: 忙啥呢，也不出来聊天。] *User info:* [Topic: Shopping, Fruit, Mango Location: Tai'an, Shandong] *Dialogue history:* [User: What are you busy with? Why don't you come out and chat?]		
Golden	*Query Type:* Empty Query; *Query*: 不需要查询 *Query Type:* Empty Query; *Query*: No need to query		
TextRank	山东省泰安市 Tai'an, Shandong	SIFRank	芒果 Mango
CatSeqTG	不需要查询 No need to query	Wr-SetTrans	不需要查询 No need to query
BART	不需要查询 No need to query	EVA2.0	不需要查询 No need to query
Dial-QP	*Keywords:* 芒果, 水果; *Predicted Query Type:* Empty Query *Predicted Query*: 不需要查询 *Keywords:* Mango, Fruit; *Predicted Query Type:* Empty Query *Predicted Query*: No need to query		

5 Conclusion

We propose a BART-based *Dial-QP* for the CQP task. Unlike previous works that only generate a dialogue query by an end-to-end paradigm, we firstly classify the query into three kinds: *Present Query*, *Absent Query* and *Empty Query*, which is a more fine-grained perspective to process the different types of queries. Subsequently, *Dial-QP* introduces a novel multitasking training paradigm to predict *No Query* to avoid the impact of this genetic query, and a *Keywords Recognition* stage to utilize the hinting keywords to guide the CQP model recognize the key information in the multi-turn conversation. Finally, experimental results have shown the very competitive performance of *Dial-QP*.

Acknowledgement. This work is supported by the Yunnan Province Education Department Foundation under the Grant No. 2023j0024.

References

1. Bao, S., et al.: Plato-xl: exploring the large-scale pre-training of dialogue generation. arXiv preprint arXiv:2109.09519 (2021)
2. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: Yake! keyword extraction from single documents using multiple local features. Inform. Sci. (2020)
3. Chen, W., Gao, Y., Zhang, J., King, I., Lyu, M.R.: -Guided encoding for keyphrase generation. In: AAAI, pp. 6268–6275 (2019)
4. Chowdhury, M.F.M., Rossiello, G., Glass, M., Mihindukulasooriya, N., Gliozzo, A.: Applying a generic sequence-to-sequence model for simple and effective keyphrase generation. arXiv preprint arXiv:2201.05302 (2022)
5. Florescu, C., Caragea, C.: A new scheme for scoring phrases in unsupervised keyphrase extraction. In: Jose, J.M., et al. (eds.) ECIR 2017. LNCS, vol. 10193, pp. 477–483. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56608-5_37
6. Florescu, C., Caragea, C.: Positionrank: an unsupervised approach to keyphrase extraction from scholarly documents. In: ACL (Volume 1: Long Papers) (2017)
7. Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., Callison-Burch, C.: Comparison of diverse decoding methods from conditional language models. In: ACL (2019)
8. Komeili, M., Shuster, K., Weston, J.: Internet-augmented dialogue generation. arXiv preprint arXiv:2107.07566 (2021)
9. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
10. Li, J., Monroe, W., Jurafsky, D.: A simple, fast diverse decoding algorithm for neural generation. arXiv preprint arXiv:1611.08562 (2016)
11. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. arXiv preprint arXiv:1704.06879 (2017)
12. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: EMNLP (2004)
13. Miller, A.H., et al.: Parlai: a dialog research software platform. arXiv preprint arXiv:1705.06476 (2017)
14. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI blog (2019)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. (2020)
16. Shuster, K., Komeili, M., Adolphs, L., Roller, S., Szlam, A., Weston, J.: Language models that seek for knowledge: modular search & generation for dialogue and prompt completion. arXiv preprint arXiv:2203.13224 (2022)
17. Sun, Y., Qiu, H., Zheng, Y., Wang, Z., Zhang, C.: Sifrank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model. IEEE Access (2020)
18. Weston, J., Shuster, K.: Blender bot 2.0: an open source chatbot that builds long-term memory and searches the internet. Facebook AI (2021)

19. Wu, H., Ma, B., Liu, W., Chen, T., Nie, D.: Fast and constrained absent keyphrase generation by prompt-based learning. In: AAAI (2022)
20. Wu, S., Li, Y., Zhang, D., Zhou, Y., Wu, Z.: Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In: ACL (2020)
21. Wu, S., Xue, P., Tao, Y., Li, Y., Wu, Z.: Select, extend, and generate: Generative knowledge selection for open-domain dialogue response generation. In: DASFAA (2023)
22. Xie, B., et al.: Wr-one2set: towards well-calibrated keyphrase generation. arXiv preprint arXiv:2211.06862 (2022)
23. Ye, J., Gui, T., Luo, Y., Xu, Y., Zhang, Q.: One2set: generating diverse keyphrases as a set. arXiv preprint arXiv:2105.11134 (2021)
24. Zhou, H., et al.: Link the world: improving open-domain conversation with dynamic spatiotemporal-aware knowledge. arXiv preprint arXiv:2206.14000 (2022)

Discourse Relation-Aware Multi-turn Dialogue Response Generation

Huijie Wang[1,2], Ruifang He[1,2(✉)], Yungang Jia[3], Jing Xu[1,2], and Bo Wang[1]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{wanghj_s,rfhe,jingxu,bo_wang}@tju.edu.cn
[2] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
[3] Tianjin Branch of National Computer Network and Information Security Management Center, Tianjin, China
jiayungang@cert.org.cn

Abstract. Multi-turn dialogue response generation aims to generate a response with consideration of the context. It is not equal to multiple single-turn dialogues due to the context dependence of response. Many existing models achieve great success for response generation, but they still struggle to model the contextual semantics of dialogue history. Sequence models have difficulties to explore the interactive relations between contextual utterances, which affects the coherence of generated responses. To solve the issue, we propose a discourse relation-aware model, which encodes the contextual utterances with a directed acyclic graph neural network (DAGNN) with constraints on dialogue-specific discourse relations to better model the intrinsic structure. Besides, we introduce an auxiliary discourse relation recognition task to enhance the model's ability of representing the context. Extensive experimental results show that our proposed model outperforms baselines.

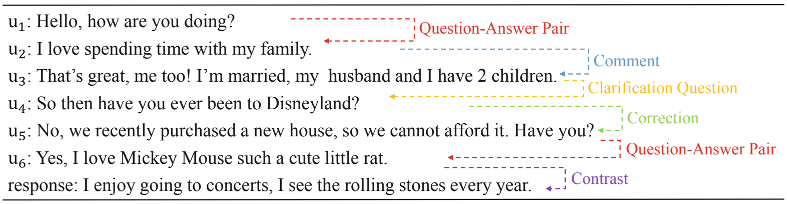
Keywords: Multi-turn Dialogue · Discourse Relation · Auxiliary Recognition Task

1 Introduction

Open-domain dialogue response generation is an important and challenging task of dialogue systems. The seq2seq models with encoder-decoder framework achieve impressive success on natural language generation (NLG), as well as the generation of open-domain dialogue response [21,22]. Most of them focus on single-turn conversational generation, which means generating response only depends on the nearest utterance. However, dialogue usually lasts for multiple rounds in real scenarios, named multi-turn dialogue.

Multi-turn dialogue response generation aims to generate a response with consideration of the context. Generating only according to the last utterance can lead incoherent and irrelevant responses. Therefore, researchers begin to explore the context information of open-domain dialogue to generate more coherent and

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 853–865, 2023.
https://doi.org/10.1007/978-3-031-44693-1_66



u_1: Hello, how are you doing?

u_2: I love spending time with my family.

u_3: That's great, me too! I'm married, my husband and I have 2 children.

u_4: So then have you ever been to Disneyland?

u_5: No, we recently purchased a new house, so we cannot afford it. Have you?

u_6: Yes, I love Mickey Mouse such a cute little rat.

response: I enjoy going to concerts, I see the rolling stones every year.

Fig. 1. An example of a multi-turn dialogue with discourse relations.

context-sensitive responses [4,19,20,23,35]. Theoretically, the number of turns of multi-turn dialogue is uncertain, which means the length of input context is unlimited. Therefore, multi-turn dialogue response generation is faced with challenges: (1) how to accurately learn the representation of context when there are many turns of dialogue history, (2) how to model the relationship between responses and context better to ensure the continuity of the dialogue.

Most existing models usually view the multi-turn context as a linear sequence of tokens [18,35]. These approaches which only utilize token-level semantics are difficult to capture relationships among utterances. Hierarchical seq2seq models, such as HRED [19] which contains hierarchical encoders, can effectively depict the structure of context. DialogBERT [4] employs a hierarchical Transformer architecture to relieve the lack of long-term information which RNN-based models suffer. But it still neglects the inherent interactive relations among utterances.

There are not only temporal relationships but also higher-level semantic relationships among utterances of context in multi-turn dialogue. As shown in Fig. 1, discourse relations are held among historical utterances in a multi-turn dialogue. Discourse relation can provide pre-defined relations between utterances [26]. Introducing the discourse relations specifical for dialogues to model context can facilitate to express the interaction between utterances, obtain more accurate contextual representations and generate more coherent responses.

To explicitly utilize discourse relations among historical utterances, inspired by DAGNN [29], we propose to model the context in the form of a directed acyclic graph (DAG). Rather than simply connecting each historical utterance, we build a DAG with constraints on discourse relations. To a certain extent, the graph structure can alleviate the impact of historical turns whose number is uncertainty or large. Besides, we introduce an auxiliary discourse relation recognition task, which aims to force the model to fully consider the relevance between context and response in the representation of context by predicting the discourse relation between the last utterance and response. We utilize BART [7] as the backbone model and enhance it with the loss of generation and classification. Extensive experiments show that our proposed dialogue discourse relation-aware BART model outperforms baseline. Our contributions can be summarized as follows:

1) Propose to model the multi-turn dialogue context in the form of DAG with constraints on implicit discourse relations between historical utterances, thus generating more context-sensitive responses.

2) Incorporate implicit discourse relation recognition as an auxiliary task in the training phase to help capture contextual information and improve the relevance of generated responses.
3) The experimental results on two public open-domain dialogue datasets demonstrate the effectiveness of our model.

2 Related Work

2.1 Multi-turn Dialogue Generation

Traditional models follow the sequential strcture. HRED [19] is a typical context-sensitive model with hierarchical framework, which contains a token-level and a context-level encoder. Noises could be introduced because previous methods process each historical utterance indiscriminately, so researchers introduce kinds of attention mechanism to measure the relevance between the response and each contextual utterances, including self-attention mechanism [32], additional topic level attention weights [31] and a static and dynamic attention framework [34].

Pre-trained language models (PLMs) are widely explored in natural language generation (NLG) in recent years [7,17,35]. PLMs-based methods can alleviate the blandness and lack of long-term contextual information which RNN-based models suffer. However, they still concatenate all the historical utterances into single text as the input, which is not particularly suitable for multi-turn dialogue response generation. This paper aims to further explore the inherent interactive relations between contextual utterances to represent context better.

2.2 Dialogue Discourse Relation

While previous work suggest that there are discourse relations among dialogue turns and they might be especially useful for generating possible next turn in open-domain dialogue [13,30], this idea is relatively unexplored. The Penn Discourse Treebank (PDTB) [15] targets formal texts like news and is not suitable for dialogues. The modified Segmented Discourse Reprsentation Theory (SDRT) hierarchy defined in STAC dataset [1] is designed specifically for dialogue, defining 16 relation types. Following SDTR, dialogue discourse relation is widely applied to various natural language processing (NLP) tasks, such as machine reading comprehension [8,9,11], meeting summarization [3], and emotion recognition [27]. In this paper, we utilize the dialogue discourse relations to help model the interactive relations between utterances in context.

3 Method

3.1 Problem Definition

Let $D = (u_1, u_2, \ldots, u_T, u_R)$ denotes a multi-turn dialogue, where the context is described as $C = (u_1, u_2, \ldots, u_T)$ and u_R is the response. Each $u_i =$

$(w_1^i, w_2^i, \ldots, w_{|u_i|}^i)$ in C is a utterance and w_j^i is the j-th word in u_i. Multi-turn dialogue response generation targets to generate a response u_R according to the given context C. To better model the interactive relations of context, we formulate a dialogue as a directed acyclic graph $G(\mathcal{V}, \mathcal{E}, \mathcal{R})$. Accordingly, the whole problem can be expressed as generating a response u_R based on the given G. Figure 2 gives the overall framework of our proposed model.

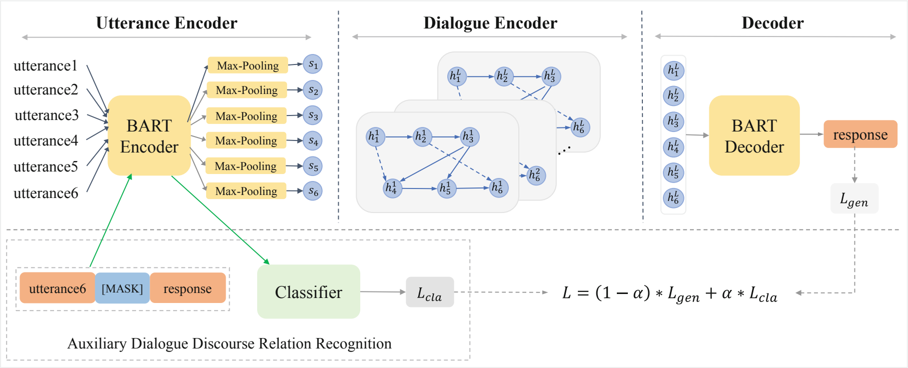


Fig. 2. The overall framework of our proposed model.

3.2 Utterance Encoder

To facilitate the understanding of context C, utterances are firstly encoded to form representations. We encode $u_i = (w_1^i, \ldots, w_{|u_i|}^i)$ with BART-Encoder, then the H_i is max-pooled to obtain the aggregated representation:

$$H_i = BART\text{-}Encoder([e_{cls}^i, e_1^i, \ldots, e_j^i, \ldots, e_{|u_i|}^i]) \tag{1}$$

$$s_i = Linear(Max\text{-}Pooling(H_i)) \tag{2}$$

where e_j^i is the embedding of the j-th word in u_i. And $s_i \in \mathbb{R}^{d_u}$ where d_u is the dimension of utterance representation.

3.3 Dialogue Encoder

Most previous models organize the entire context as a sequence, which falls short for capturing the interactive relations of context. We design a DAG and incorporate the dialogue discourse relations to model the context better.

Dialogue Discourse Relation. Following [1], there are sixteen discourse relations in total, including comment, clarification question, elaboration, acknowledgment, continuation, explanation, conditional, question-answer pair, alternation, question elaboration, result, background, narration, correction, parallel and contrast. We utilize discourse relations to model the multi-turn dialogue context in the form of DAG, thus facilitating the model to capture the interactive relations among utterances and obtain more accurate contextual representation.

Dialogue DAGNN. In this paper, a DAG is denoted by $G(\mathcal{V}, \mathcal{E}, \mathcal{R})$, where V is a set of vertices that represents utterances and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of directed edges. The edges are formulated as $(i, j, r_{ij}) \in \mathcal{E}$, where $r_{ij} \in \mathcal{R}$ is the dialogue discourse relation type of the edge. The set $\mathcal{R}$ contains the sixteen dialogue discourse relations as mentioned above. We impose the direction constraint that a previous utterance can pass message to a future utterance, but a future utterance cannot pass message backwards. This setting is more reasonable for dialogue scene, as a utterance should not be influenced by the future utterances.

At each layer, DAGNN aggregates information for each node in temporal order and allows all nodes to gather information from neighbors and update their states at the same layer:

$$h_i^l = Comb(h_i^{l-1}, Aggre(\{h_j^l | j \in N_i\})) \tag{3}$$

where $Comb(\cdot)$ and $Aggre(\cdot)$ are parameterized neural networks for information processing and information aggregation from neighboring nodes respectively. N_i denotes the neighbours of the i-th node. We initialize $h_i^0 = s_i$ with the input representation of the i-th utterance. The hidden state of utterances are computed recurrently from the first utterance to the last one.

For each u_i at the l-th layer, the edge weights from its predecessors u_j can be calculated by using the h_i^{l-1} to attend to the predecessors' hidden states:

$$\alpha_{ij}^l = softmax_{j \in N_i}(W_\alpha^l [h_j^l || h_i^{l-1}]) \tag{4}$$

where h_i^{l-1} is the node representation at the $(l-1)$-th layer, h_j^l is the neighboring node representation that has been processed in the l-th layer. W_α^l is trainable parameters and $||$ denotes the concatenation operation. Furthermore, the aggregated message can be computed according to α_{ij}^l, and $W_{r_{ij}}^l$ is trainable parameters for discourse relation-aware transformation:

$$msg_i^l = \sum_{j \in N_i} \alpha_{ij}^l W_{r_{ij}}^l h_j^l \tag{5}$$

Then the final hidden state nod_i^l is obtained with a gated recurrent unit (GRU):

$$nod_i^l = GRU_n^l(h_i^{l-1}, msg_i^l) \tag{6}$$

Inspired by [24], msg_i^l is only used to control the propagation of u_i's hidden state, neglecting the contextual information. Therefore, another GRU is utilized to capture the contextual message ctx_i^l, and the roles of h_i^{l-1} and msg_i^l are reversed:

$$ctx_i^l = GRU_c^l(msg_i^l, h_i^{l-1}) \tag{7}$$

Finally, the node representation of utterance u_i at the l-th layer is updated by summing the two types of information:

$$h_i^l = nod_i^l + ctx_i^l \tag{8}$$

3.4 Auxiliary Discourse Relation Recognition

To facilitate the model to obtain more accurate contextual representations and consider the relation between context and response, we introduce an auxiliary task to force the model to recognize the discourse relation between utterance u_T and target response u_R. We regard u_T and u_R as two discourse units (DUs), and firstly transform them to a single string S with a special token $[MASK]$, then sent it into the encoder of BART to obtain the hidden states $\widetilde{H}$:

$$S = u_T[MASK]u_R \tag{9}$$

$$\widetilde{H} = BART\text{-}Encoder(S) \tag{10}$$

We employ the hidden states of $[MASK]$ for representing the dialogue discourse relation between two DUs. Then we feed the output $\widetilde{H}_{MASK}$ into a linear layer with GELU activation followed by a softmax layer to classify the discourse relation r between two DUs. $P(r)$ represents probability distribution of r:

$$P(r) = softmax(Linear(\widetilde{H}_{MASK})) \tag{11}$$

3.5 Generation Module

As shown in Fig. 2, we take the concatenation of h_i at the last layer as the final representation of context H_C, and send it to the decoder of BART:

$$H_C = ||_{i=1}^{T} h_i^L \tag{12}$$

$$P(w_t^R) = BART\text{-}Decoder(H_C, w_{<t}^R) \tag{13}$$

where $||$ denotes the concatenation operation. T is the length of context, and w_t^R represents the t-th word of generated response.

3.6 Training and Loss Functions

The training loss of the model consists of two parts: generation loss L_{gen} and classification loss L_{cla}, both of which employ the cross-entropy loss. The final loss L is a weighted sum of two components, and the sum of their weights equals one. The objective function is as follows, where α is an adjustable hyper-parameter:

$$L = (1 - \alpha) * L_{gen} + \alpha * L_{cla} \tag{14}$$

4 Experimental Settings

4.1 Dataset and Implementation Details

We evaluate our model with two public multi-turn dialogue datasets. The first one is PersonaChat [33], which provides natural conversations between two participants. The second is Ubuntu Dialogue Corpus [12]. It contains dialogues collected from the Ubuntu Internet Relayed Chat channel, each of which is about

Table 1. Statistics of two datasets annotated with dialogue discourse relations.

	PersonaChat			Ubuntu		
	Train	Valid	Test	Train	Valid	Test
Number of dialogues	16738	1000	1000	162496	6600	6600
Average number of turns	13.65	14.56	13.58	7.69	7.49	7.59
Average number of relations	12.65	13.56	12.58	6.69	6.49	6.59

a Ubunturelated technical problem. We randomly split two datasets to training, validation and test sets respectively. To obtain discourse relations, we firstly employ a dialogue discourse relation parser [25]. Table 1 shows detailed statistics of two datasets. We filter the datasets for dialogues with less than 5 utterances, which is more consistent with multi-turn scenarios.

We implement all the models using PyTorch, and conduct hyper-parameter search for all models. There are 6 layers for encoder and decoder in our model, which has 768 dimensional embedding size and hidden states. For the α mentioned earlier, we let $\alpha = 0.3$ for the overall performance comparison by default. During the training phase, we set the learning rate as 1e-5 and optimized with Adam optimizer [6]. Checkpoints with the top performance are finally evaluated on the test set to report final results.

4.2 Baselines

We select the following baseline methods for comparison: (1) **Seq2Seq** [28] is a standard RNN-based model with attention mechanism. (2) **HRED** [19] is a typical generation model for multi-turn dialogue, which firstly introduces a hierarchical encoder-decoder framework. (3) **ReCoSa** [32] is a RNN-based model with self-attention mechanism. It is proposed to discriminate the importance of contextual utterances. (4) **BART** [7] is a pre-trained Transformer-based model for conditional text generation. It consists of a bidirectional encoder and a left-to-right decoder. (5) **DialoGPT** [35] is a pre-trained model for dialogue generation. It is pre-trained based on GPT-2[17] using Reddit comments. (6) **Dialog-BERT** [4] is a pre-trained dialogue generation model. It employs a hierarchical Transformer architecture to represent the context. (7) **DialogVED** [2] is a new dialogue pre-trained framework, which introduces continuous latent variables to increase the diversity of generated responses.

What's more, we perform ablation studies to validate the effectiveness of each part of our model: (1) **-recognition** is the ablated model which removes the auxiliary discourse relation recognition task. (2) **-relations** removes DAGNN and simply connects each historical utterance without discourse relations.

4.3 Metric

For evaluation, we employ the automatic metrics and human metrics to evaluate the quality of generated responses. We adopt widely used **BLEU-n** [14] to

Table 2. Results of automatic evaluation. The best results are highlighted in bold.

Model	PersonaChat				Ubuntu			
	BLEU-1	BLEU-2	Dist-1	Dist-2	BLEU-1	BLEU-2	Dist-1	Dist-2
Seq2Seq	15.13	8.91	0.0086	0.0251	4.32	2.87	0.0092	0.0322
HRED	20.91	12.94	0.0197	0.0556	9.39	5.82	0.0196	0.0727
ReCoSa	23.66	13.42	0.0445	0.1807	12.77	7.54	0.0322	0.0898
BART	27.68	15.25	0.1408	0.449	15.22	8.78	0.0644	0.4278
DialoGPT	26.82	15.43	0.1358	0.541	15.83	8.62	0.0694	0.4596
DialogBERT	27.41	16.58	0.1382	**0.5713**	16.77	9.17	0.0544	0.4278
DialogVED	29.17	**20.6**	0.0957	0.3938	15.66	8.32	0.0653	0.4955
Ours	**30.34**	18.52	**0.148**	0.4625	**18.85**	**11.75**	**0.0755**	**0.515**
-recognition	29.49	16.48	0.1360	0.4489	17.56	10.56	0.0691	0.431
-relations	28.07	15.75	0.1316	0.4461	16.736	9.43	0.0629	0.402

measure n-grams overlapping proportion, which can evaluate the relevance of the generated response to reference. Besides, to evaluate the model's capacity on lexical diversity, we utilize the corpus-level metric of **distinct-1/2** [10]. It is defined as the number of distinct uni- or bi-grams divided by the total number of words, which measures the proportion of unique n-grams in generated responses.

Only considering automatic metrics may not suitably reflect the quality of generation, we further evaluate with manual metrics: (1) **Coherence** measures whether the response is relevant with the context. The score ranges from 1 to 5, higher score means more coherent. (2) **Informativeness** evaluates which response conveys more information, the score ranges from 1 to 5, higher score means more informative. (3) **Sensibleness** measures whether the model can capture the dialogue discourse relations between utterances. The score ranges from 1 to 5, higher score means the generated response is more sensitive to dialogue discourse relations.

5 Results and Analysis

5.1 Automatic Evaluation

We present the overall results of all the compared methods at Table 2. Generally, our proposed model outperforms baselines on most metrics, which shows its effectiveness for multi-turn dialogue response generation. For baselines, pre-trained models achieve higher scores than traditional RNN-based models, especially in terms of distinct measures. It demonstrates the strong ability of models with numerous data pre-trained to generate diverse text. Comparing Seq2Seq and DialoGPT with HRED and DialogBERT respectively, we can observe that models with hierarchical framework perform better than non-hierarchical models, which proves that mining the structural information of dialogue history is beneficial to improve the quality of generation.

Our model obtains higher scores than BART, which verifies that introducing graph structure and dialogue discourse relation to model the context helps multi-turn dialogue response generation. Compared with DialogVED, our model cannot reach the best on some metrics. We speculate it is because DialogVED adopts the future predicting strategy in ProphetNet [16] when decoding, and introduces latent variables to help generation procedure. However, the large parameter scale of DialogVED (392M) leads to overfitting when fine-tuning on the datasets. What's more, it takes the entire context as a long text input so that it can't make full use of the interaction information among historical utterances.

5.2 Human Evaluation

Considering automatic metrics may not suitably reflect the effectiveness, we further perform human evaluation. We randomly sample 50 testing dialogues and corresponding generated responses of each models from two datasets respectively. Three participants are required to range the generations from different perspectives and we report their average scores. As presented at Table 3, hierarchical models perform better than non-hierarchical models in terms of coherence and our model gets the highest score. It validates the effectiveness of introducing graph structure and dialogue discourse relation to model the context. Early pretrained models perform generally in sensibleness, we suppose it is because they cannot absorb and understand given context as [5] point. In general, our model can generate context-sensible responses, and we further analyze in case study.

Table 3. Results of human evaluation.

	Metric	Seq2Seq	HRED	BART	DialogVED	Ours
PersonaChat	coherence	1.81	3.15	2.92	3.95	4.17
	informativeness	2.19	2.23	3.25	3.71	3.62
	sensibleness	1.55	2.04	2.31	3.54	4.33
Ubuntu	coherence	1.45	2.87	3.32	3.89	4.31
	informativeness	1.78	2.21	3.56	4.21	3.97
	sensibleness	1.39	2.44	3.11	3.67	4.42

5.3 Ablation Study

We conducted ablation studies to verify the effectiveness of each component in our model. As shown in Table 2, the ablated models cannot catch up to the complete model, which validates discourse relation-aware module and graph structure are effective for multi-turn dialogue response generation.

Without auxiliary discourse relation recognition task, the -recognition gets slightly lower score than the complete model, which indicates that it has facilitating effect on modeling context dependence to some extent. The performance of the -relations lags behind the complete model as shown, which indicates that discarding the directed acyclic graph structure and ignoring the dialogue discourse relations between historical uterrances have large impact on the overall performance. The -relation cannot adequately mine the contextual information.

5.4 Effect of Dialogue Discourse Relation

Consistent results are obtained in both automatic and human evaluation, proving the importance of discourse relation-aware module. It's worth noting that we employ a dialogue discourse relation parser [25] to annotate datasets. The parser's accuracy of implicit discourse relation recognition is affected by many factors. And the discourse relation recognition is a multi-label task [8], there could be more than one relation between two utterances, which would easily make ambiguities. So the accuracy of the parser is always limited, which can probably introduce noises when modeling context in the form of DAG with constraints on the annotated discourse relations.

5.5 Case Study

Table 4 shows comparisons on PersonaChat of our model and four main baselines. The sample illustrates that our model generates more context-relevant responses than baseline models. The result of BART is less relevant to the context. We suppose that it is because BART is a pure auto-regressive language model which takes the entire context as a long text input. The generation of DialogVED looks similar to the golden response. But our model generates more semantically consistent response and it fits the dialogue discourse relation better.

Table 4. Case analysis on PersonaChat. **Continuation** repesents the dialogue discourse relation between response and the last utterance.

Context	u_1: Good morning, I hope you are having a great day so far
	u_2: Good morning, it's going pretty well how is your day?
	u_3: Just getting started really. I am not much of a morning person so coffee coffee
	u_4: I understand that one. I am gonna play a new video game I got yesterday
	u_5: Oh sweet what did you get?
	u_6: Destiny 2 I loved the first one
	u_7: All my people are gamers so I hear a lot about them even though I do not play
	u_8: My family is the same. What do you do for a living?
	u_9: I resell jewelry. I just got back from a show where I got new hoop earrings
GOLD	That sounds like a rewarding job. I work in marketing. (**Continuation**)
Seq2Seq	That's true. <unk> is my favorite type of car now
HRED	Wow that sounds interesting. I love music
BART	Hooprings are really cool I love those
DialogVED	That sounds like a lot of fun
Ours	**That's cool. What kind of jewelry do you sell?**

6 Conclusion

In this paper, we propose a discourse relation-aware model for the generation of open-domain multi-turn dialogue. Discourse relation plays an important role in understanding dialogue history, so we employ a DAGNN to model the context with constraints on discourse relations. Meanwhile, we introduce an auxiliary discourse relation recognition task to force the model to consider the contextual dependence in the representations of context. The experimental results demonstrate the effectiveness of our method.

Acknowledgement. Our work is supported by the National Natural Science Foundation of China under Grant (61976154).

References

1. Asher, N., Hunter, J., Morey, M., Benamara, F., Afantenos, S.: Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In: Proceedings of the 10th LREC, pp. 2721–2727 (2016)
2. Chen, W., et al.: Dialogved: a pre-trained latent variable encoder-decoder model for dialog response generation. In: Proceedings of the 60th ACL, pp. 4852–4864 (2022)
3. Feng, X., Feng, X., Qin, B., Geng, X.: Dialogue discourse-aware graph model and data augmentation for meeting summarization. In: Proceedings of the IJCAI, pp. 3808–3814 (2021)
4. Gu, X., Yoo, K.M., Ha, J.W.: Dialogbert: discourse-aware response generation via learning to recover and rank utterances. In: Proceedings of the AAAI, pp. 12911–12919 (2021)
5. Gubelmann, R., Handschuh, S.: Context matters: a pragmatic study of plms' negation understanding. In: Proceedings of the 60th ACL, pp. 4602–4621 (2022)
6. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th ACL, pp. 7871–7880 (2020)
8. Li, J., Liu, M., Kan, M.Y., Zheng, Z., Wang, Z., et al.: Molweni: a challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In: Proceedings of the 28th COLING, pp. 2642–2652 (2020)
9. Li, J., et al.: Dadgraph: a discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension. In: Proceedings of the 2021 IJCNN, pp. 1–8 (2021)
10. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, W.B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of the 2016 NAACL: Human Language Technologies, pp. 110–119 (2016)
11. Li, Y., Zhao, H.: Self-and pseudo-self-supervised prediction of speaker and key-utterance for multi-party dialogue reading comprehension. In: Findings of the 2021 EMNLP, pp. 2053–2063 (2021)
12. Lowe, R., Pow, N., Serban, I.V., Pineau, J.: The ubuntu dialogue corpus: a large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the 16th SIGDIAL, pp. 285–294 (2015)

13. Ma, M.D., Bowden, K., Wu, J., et al.: Implicit discourse relation identification for open-domain dialogues. In: Proceedings of the 57th ACL, pp. 666–672 (2019)
14. Papineni, K., Roukos, S., Ward, T., et al.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th ACL, pp. 311–318 (2002)
15. Prasad, R., et al.: The penn discourse treebank 2.0. In: Proceedings of the 6th LREC (2008)
16. Qi, W., et al.: Prophetnet: predicting future n-gram for sequence-to-sequencepre-training. In: Findings of the 2020 EMNLP, pp. 2401–2410 (2020)
17. Radford, A., et al.: Language models are unsupervised multitask learners. In: OpenAI blog, p. 9 (2019)
18. Roller, S., et al.: Recipes for building an open-domain chatbot. In: Proceedings of the EACL (2021)
19. Serban, I., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: Proceedings of the AAAI, pp. 3776–3783 (2016)
20. Serban, I., et al.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: Proceedings of the AAAI, pp. 3295–3301 (2017)
21. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: Proceedings of the 53rd ACL and the 7th IJCNLP, pp. 1577–1586 (2015)
22. Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., Kurzweil, R.: Generating high-quality and informative conversation responses with sequence-to-sequence models. In: Proceedings of the 2017 EMNLP, pp. 2210–2219 (2017)
23. Shen, L., Feng, Y., Zhan, H.: Modeling semantic relationship in multi-turn conversations with hierarchical latent variables. In: Proceedings of the 57th ACL, pp. 5497–5502 (2019)
24. Shen, W., Wu, S., Yang, Y., Quan, X.: Directed acyclic graph network for conversational emotion recognition. In: Proceedings of the 59th ACL and the 11th IJCNLP, pp. 1551–1560 (2021)
25. Shi, Z., Huang, M.: A deep sequential model for discourse parsing on multi-party dialogues. In: Proceedings of the AAAI, pp. 7007–7014 (2019)
26. Stone, M., Stojnic, U., Lepore, E.: Situated utterances and discourse relations. In: Proceedings of the 10th IWCS, pp. 390–396 (2013)
27. Sun, Y., Yu, N., Fu, G.: A discourse-aware graph neural network for emotion recognition in multi-party conversation. In: Findings of the 2021 EMNLP, pp. 2949–2958 (2021)
28. Sutskever, I., et al.: Sequence to sequence learning with neural networks. In: Proceedings of the NeurIPS, vol. 195, pp. 496–527 (2014)
29. Thost, V., Chen, J.: Directed acyclic graph neural networks. In: Proceedings of the ICLR (2021)
30. Tonelli, S., Riccardi, G., Prasad, R., Joshi, A.: Annotation of discourse relations for conversational spoken dialogs. In: Proceedings of the 7th LREC (2010)
31. Zhang, H., Lan, Y., Pang, L., Chen, H., et al.: Modeling topical relevance for multi-turn dialogue generation. In: Proceedings of the IJCAI, pp. 3737–3743 (2020)
32. Zhang, H., Lan, Y., Pang, L., Guo, J., Cheng, X.: Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In: Proceedings of the 57th ACL, pp. 3721–3730 (2019)
33. Zhang, S., Dinan, E., Urbanek, J., et al.: Personalizing dialogue agents: I have a dog, do you have pets too? In: Proceedings of the 56th ACL, pp. 2204–2213 (2018)

34. Zhang, W., et al.: A static and dynamic attention framework for multi turn dialogue generation. In: JACM, vol. 41 (2022)
35. Zhang, Y., et al.: Dialogpt: large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th ACL: System Demonstrations, pp. 270–278 (2020)

Author Index

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 867–875, 2023.
https://doi.org/10.1007/978-3-031-44693-1