LNAI 14326

Fenrong Liu · Arun Anand Sadanandan ·
Duc Nghia Pham · Petrus Mursanto ·
Dickson Lukose (Eds.)

PRICAI 2023: Trends in Artificial Intelligence

20th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2023
Jakarta, Indonesia, November 15–19, 2023
Proceedings, Part II

2 Part II

PRICAI

Springer

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence 14326

Founding Editor

Jörg Siekmann

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.

Fenrong Liu · Arun Anand Sadanandan ·
Duc Nghia Pham · Petrus Mursanto ·
Dickson Lukose
Editors

PRICAI 2023: Trends in Artificial Intelligence

20th Pacific Rim
International Conference on Artificial Intelligence, PRICAI 2023
Jakarta, Indonesia, November 15–19, 2023
Proceedings, Part II

Springer

Editors

Fenrong Liu
Tsinghua University
Beijing, China

Duc Nghia Pham
MIMOS (Malaysia)
Kuala Lumpur, Malaysia

Dickson Lukose
Tabcorp Holdings Limited
Melbourne, VIC, Australia

Arun Anand Sadanandan
SEEK Limited
Cremorne, NSW, Australia

Petrus Mursanto
Universitas Indonesia
Depok, Indonesia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-981-99-7021-6 ISBN 978-981-99-7022-3 (eBook)
https://doi.org/10.1007/978-981-99-7022-3

LNCS Sublibrary: SL7 – Artificial Intelligence

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

Preface

Greetings and welcome to 20th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2023). It was an honor to convene this significant event in a hybrid format in Jakarta, Indonesia. It was indeed a privilege for the Faculty of Computer Science at Universitas Indonesia to undertake the role of hosting these pivotal discussions that reach beyond the academic realm, advancing societies and economies across the Pacific Rim and Oceania.

This year, we received a remarkable 422 submissions: 354 for the Main track and 68 for the AI-Impact track. Every submission underwent a rigorous double-blind review process, receiving a minimum of 3 reviews, and in some cases up to 6. Throughout the process, the program committee (PC) members engaged in discussions, with additional reviews sourced as needed, prior to finalizing recommendations. The program chairs then assessed the reviews and comments, calibrating discrepancies in individual reviews and ratings to maintain decision consistency. The collective effort of the entire program committee, including chairs, 409 PC members, and 91 external reviewers, was monumental in ensuring a fair and consistent selection process. We ultimately accepted 95 regular papers and 36 short papers for oral presentation, resulting in a 22.51% acceptance rate for regular papers and an overall acceptance rate of 31.04%. Additionally, a comprehensive quality control procedure was introduced for camera-ready papers. The aim was to prompt authors to incorporate the feedback provided by PC members and reviewers into their final submissions. Content similarity checks were also performed to ensure that the similarity rate did not exceed 15%.

The technical program was comprehensive and intellectually engaging, featuring five workshops, nine tutorials, two panel discussions, and the main conference sessions. All regular and short papers were orally presented over three days in parallel and in topical program sessions. We were honored to have some of the brightest minds in AI to share their insights and enrich our collective understanding: Thomas Anton Kochan (Massachusetts Institute of Technology, USA), Hanna Kurniawati (Australian National University, Australia), Anand Rao (Carnegie Mellon University, USA), and Geoff Webb (Monash University, Australia).

A heartfelt thanks was expressed towards the organizing committee for their tireless and unwavering efforts that facilitated the success of this event. A special recognition to Adila Alfa Krisnadhi for his leadership on local arrangements. We would also like to acknowledge our workshop and tutorial organizers, who formed the core of our technical program. These dedicated individuals brought a diverse range of expertise that promised to deepen our exploration of AI technologies.

We would like to thank our advisory board members for their invaluable guidance during the planning stages. A special recognition to Abdul Sattar for his extraordinary contribution towards planning, execution, and a conference site visit that contributed to the success of PRICAI 2023. Furthermore, we extend our gratitude to the PRICAI Steering Committee for entrusting us with the privilege of hosting this impactful conference.

We would not have been here without the support of our sponsors, whose commitment enabled us to keep pushing boundaries. To them, as well as all participants in this event, thank you.

As we delved into the various topics that PRICAI 2023 had to offer, let us remind ourselves that our deliberations have a lasting impact on the future of AI in the Pacific Rim and beyond. We genuinely hope that our time spent at PRICAI 2023 will pave the way for innovations that are both groundbreaking and beneficial.

November 2023

Fenrong Liu
Arun Anand Sadanandan
Duc Nghia Pham
Dickson Lukose
Petrus Mursanto

Organization

PRICAI Steering Committee

Steering Committee

Quan Bai	University of Tasmania, Australia
Tru Hoang Cao	University of Texas Health Science Center at Houston, USA
Xin Geng	Southeast University, China
Guido Governatori	Reasoning Research Institute, Australia
Takayuki Ito	Kyoto University, Japan
Byeong-Ho Kang	University of Tasmania, Australia
M. G. M. Khan	University of the South Pacific, Fiji
Sankalp Khanna	CSIRO Australian e-Health Research Centre, Australia
Fenrong Liu	Tsinghua University, China
Dickson Lukose	Tabcorp Holdings Ltd., Australia
Hideyuki Nakashima	Sapporo City University, Japan
Abhaya Nayak	Macquarie University, Australia
Seong Bae Park	Kyung Hee University, South Korea
Duc Nghia Pham	MIMOS Berhad, Malaysia
Abdul Sattar	Griffith University, Australia
Alok Sharma	RIKEN, Japan & University of the South Pacific, Fiji
Thanaruk Theeramunkong	Thammasat University, Thailand
Zhi-Hua Zhou	Nanjing University, China

Honorary Members

Randy Goebel	University of Alberta, Canada
Tu-Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Mitsuru Ishizuka	University of Tokyo, Japan
Hiroshi Motoda	Osaka University, Japan
Geoff Webb	Monash University, Australia
Albert Yeap	Auckland University of Technology, New Zealand
Byoung-Tak Zhang	Seoul National University, South Korea
Chengqi Zhang	University of Technology Sydney, Australia

Conference Organizing Committee

General Chairs

Dickson Lukose	Tabcorp Holdings Ltd., Australia
Petrus Mursanto	Universitas Indonesia, Indonesia

Program Chairs

Fenrong Liu	Tsinghua University, China
Arun Anand Sadanandan	SEEK, Australia
Duc Nghia Pham	MIMOS Berhad, Malaysia

Local Organizing Chair

Adila Alfa Krisnadhi	Universitas Indonesia, Indonesia

Workshop Chairs

Evi Yulianti	Universitas Indonesia, Indonesia
Takahiro Uchiya	Nagoya Institute of Technology, Japan

Tutorial Chairs

Fariz Darari	Universitas Indonesia, Indonesia
M. A. Hakim Newton	University of Newcastle, Australia

Publicity Chairs

Panca Hadi Putra	Universitas Indonesia, Indonesia
Md Khaled Ben Islam	Griffith University, Australia

Advisory Board

Abdul Sattar	Griffith University, Australia
Hammam Riza	KORIKA; University of Syiah Kuala, Indonesia
Patricia Anthony	Lincoln University, New Zealand
Jirapun Daengdej	Merlin's Solutions International, Thailand
Seong Bae Park	Kyung Hee University, South Korea
M. G. M. Khan	University of the South Pacific, Fiji

Qingliang Chen	Jinan University, China
Takayuki Ito	Kyoto University, Japan
Tru Hoang Cao	University of Texas Health Science Center at Houston, USA
Sankalp Khanna	CSIRO Australian e-Health Research Centre, Australia
Stéphane Bressan	National University of Singapore, Singapore
Hideyuki Nakashima	Sapporo City University, Japan

Program Committee

Tooba Aamir	Data61, CSIRO, Australia
Azizi Ab Aziz	Universiti Utara Malaysia, Malaysia
Taufik Abidin	Universitas Syiah Kuala, Indonesia
Kiki Adhinugraha	La Trobe University, Australia
Martin Aleksandrov	Freie Universität Berlin, Germany
Hissah Alotaibi	University of Melbourne, Australia
Sagaya Amalathas	University of Southampton, Malaysia
Galia Angelova	Bulgarian Academy of Sciences, Bulgaria
Patricia Anthony	Lincoln University, New Zealand
Ryuta Arisaka	Kyoto University, Japan
Mohammad Arshi Saloot	MIMOS Berhad, Malaysia
Siti Liyana Azman	International Islamic University Malaysia, Malaysia
Mohamed Jaward Bah	Zhejiang Lab, China
Quan Bai	University of Tasmania, Australia
Thirunavukarasu Balasubramaniam	Queensland University of Technology, Australia
Arishnil Kumar Bali	University of the South Pacific, Fiji
Vishnu Monn Baskaran	Monash University, Malaysia
Chutima Beokhaimook	Rangsit University, Thailand
Pascal Bercher	Australian National University, Australia
Ateet Bhalla	Independent Technology Consultant, India
Hanif Bhuiyan	Monash University, Australia
Ran Bi	Dalian University of Technology, China
Thomas Bolander	Technical University of Denmark, Denmark
Chih How Bong	Universiti Malaysia Sarawak, Malaysia
Aida Brankovic	CSIRO, Australia
Chenyang Bu	Hefei University of Technology, China
Agus Buono	Bogor Agriculture University, Indonesia
Xiongcai Cai	University of New South Wales, Australia

Jian Cao	Shanghai Jiao Tong University, China
Tru Cao	University of Texas Health Science Center at Houston, USA
Sixian Chan	Zhejiang University of Technology, China
Narayan Changder	National Institute of Technology Durgapur, India
Hutchatai Chanlekha	Kasetsart University, Thailand
Kaylash Chaudhary	University of the South Pacific, Fiji
Bincai Chen	Dalian University of Technology, China
Gang Chen	Victoria University of Wellington, New Zealand
Liangyu Chen	East China Normal University, China
Qi Chen	Victoria University of Wellington, New Zealand
Rui Chen	Nankai University, China
Siqi Chen	Tianjin University, China
Songcan Chen	Nanjing University of Aeronautics and Astronautics, China
Tingxuan Chen	Central South University, China
Weitong Chen	University of Adelaide, Australia
Weiwei Chen	Sun Yat-sen University, China
Wu Chen	Southwest University, China
Yakun Chen	University of Technology Sydney, Australia
Yingke Chen	Northumbria University, UK
Wai Khuen Cheng	Universiti Tunku Abdul Rahman, Malaysia
Yihang Cheng	Tianjin University, China
Boonthida Chiraratanasopha	Yala Rajabhat University, Thailand
Cody Christopher	Data61, CSIRO, Australia
Jinmiao Cong	Dalian University of Technology, China
Dan Corbett	University of Sydney, Australia
Zhihong Cui	Shandong University, China
Jirapun Daengdej	Assumption University of Thailand, Thailand
Li Dai	Zaozhuang University, China
Fariz Darari	Universitas Indonesia, Indonesia
Iman Dehzangi	Rutgers University, USA
Zelin Deng	Changsha University of Science and Technology, China
Chandra Kusuma Dewa	Universitas Islam Indonesia, Indonesia
Sarinder Kaur Dhillon	Universiti Malaya, Malaysia
Shiyao Ding	Kyoto University, Japan
Zheng Dong	Baidu, China
Shyamala Doraisamy	University Putra Malaysia, Malaysia
Ellouze Ellouze	University of Sfax, Tunisia
Uzoamaka Ezeakunne	Florida State University, USA
Lei Fan	University of New South Wales, Australia

Chastine Fatichah	Institut Teknologi Sepuluh Nopember, Indonesia
Shanshan Feng	Shandong Normal University, China
Xiao Feng	University of Electronic Science and Technology of China, China
Valnir Ferreira Jr.	Independent Consultant, Australia
Muhammad Firoz-Mridha	American International University-Bangladesh, Bangladesh
Tim French	University of Western Australia, Australia
Xiaoxuan Fu	China University of Political Science and Law, China
Somchart Fugkeaw	Thammasat University, Thailand
Katsuhide Fujita	Tokyo University of Agriculture and Technology, Japan
Naoki Fukuta	Shizuoka University, Japan
Hua Leong Fwa	Singapore Management University, Singapore
Marcus Gallagher	University of Queensland, Australia
Dragan Gamberger	Ruđer Bošković Institute, Croatia
Jian Gao	Northeast Normal University, China
Xiaoying Gao	Victoria University of Wellington, New Zealand
Xin Geng	Southeast University, China
Yasmeen George	Monash University, Australia
Sujata Ghosh	Indian Statistical Institute, India
Michael Granitzer	University of Passau, Germany
Alban Grastien	Australian National University, Australia
Charles Gretton	Australian National University, Australia
Wen Gu	Japan Advanced Institute of Science and Technology, Japan
Jiawei Guo	Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
Avisek Gupta	TCG CREST, India
Fikret Gurgen	Boğaziçi University, Turkey
Julian Gutierrez	Monash University, Australia
Rafik Hadfi	Kyoto University, Japan
Misgina Tsighe Hagos	University College Dublin, Ireland
Mourad Hakem	Université de Franche-Comté, France
Bavly Hanna	University of Technology Sydney, Australia
Jawad Ahmad Haqbeen	Kyoto University, Japan
Md Mahmudul Hasan	University of New South Wales, Australia
Mehedi Hasan	BRAC University, Bangladesh
David Hason Rudd	University of Technology Sydney, Australia
Hamed Hassanzadeh	CSIRO, Australia
Tessai Hayama	Nagaoka University of Technology, Japan

Priyanto Hidayatullah	Politeknik Negeri Bandung, Indonesia
Linlin Hou	Zhejiang Lab, China
Shuyue Hu	Shanghai Artificial Intelligence Laboratory, China
Jiwei Huang	China University of Petroleum, China
Victoria Huang	National Institute of Water and Atmospheric Research, New Zealand
Xiaodi Huang	Charles Sturt University, Australia
Nguyen Duy Hung	Thammasat University, Thailand
Huan Huo	University of Technology Sydney, Australia
Habibi Husain Arifin	Assumption University of Thailand, Thailand
Du Huynh	University of Western Australia, Australia
Van Nam Huynh	Japan Advanced Institute of Science and Technology, Japan
Masashi Inoue	Tohoku Institute of Technology, Japan
Md Khaled Ben Islam	Griffith University, Australia
Md. Saiful Islam	University of Newcastle, Australia
Takayuki Ito	Kyoto University, Japan
Sanjay Jain	National University of Singapore, Singapore
Mehrdad Jalali	Karlsruhe Institute of Technology, Germany
Fatemeh Jalalvand	Data61, CSIRO, Australia
Wojtek Jamroga	Polish Academy of Sciences, Poland
Wisnu Jatmiko	Universitas Indonesia, Indonesia
Jingjing Ji	Huazhong University of Science and Technology, China
Liu Jiahao	Southwest University, China
Guifei Jiang	Nankai University, China
Jianhua Jiang	Jilin University of Finance and Economics, China
Ting Jiang	Zhejiang Lab, China
Yuncheng Jiang	South China Normal University, China
Nattagit Jiteurtragool	King Mongkut's University of Technology North Bangkok, Thailand
Rui-Yang Ju	Tamkang University, Taiwan
Iman Kamkar	Deloitte, Australia
Hideaki Kanai	Japan Advanced Institute of Science and Technology, Japan
Rathimala Kannan	Multimedia University, Malaysia
Natsuda Kaothanthong	Thammasat University, Thailand
Jessada Karnjana	National Electronics and Computer Technology Center, Thailand
Shohei Kato	Nagoya Institute of Technology, Japan
Natthawut Kertkeidkachorn	Japan Advanced Institute of Science and Technology, Japan
Nor Khalid	Universiti Teknologi MARA, Malaysia

Jane Jean Kiam	Universität der Bundeswehr München, Germany
Huan Koh	Monash University, Australia
Kazunori Komatani	Osaka University, Japan
Sébastien Konieczny	French National Centre for Scientific Research, France
Harindu Korala	Monash University, Australia
Fajri Koto	Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
Adila A. Krisnadhi	Universitas Indonesia, Indonesia
Alfred Krzywicki	University of Adelaide, Australia
Charles Kuan	Tabcorp Holdings Limited, Australia
Li Kuang	Central South University, China
Dinesh Kumar	University of the South Pacific, Fiji
Shiu Kumar	Fiji National University, Fiji
Young-Bin Kwon	Chung-Ang University, South Korea
Ho-Pun Lam	Independent Researcher, Australia
Davide Lanti	Free University of Bozen-Bolzano, Italy
Roberto Legaspi	KDDI Research, Japan
Dazhu Li	Chinese Academy of Sciences, China
Gang Li	Deakin University, Australia
Guangliang Li	Ocean University of China, China
Guoqiang Li	Shanghai Jiao Tong University, China
Ren Li	Chongqing Jiaotong University, China
Tianrui Li	Southwest Jiaotong University, China
Weihua Li	Auckland University of Technology, New Zealand
Yicong Li	University of Technology Sydney, Australia
Yuan-Fang Li	Monash University, Australia
Xiubo Liang	Zhejiang University, China
Ariel Liebman	Monash University, Australia
Alan Wee-Chung Liew	Griffith University, Australia
Donghui Lin	Okayama University, Japan
Chanjuan Liu	Dalian University of Technology, China
Di Liu	Inner Mongolia University, China
Fenrong Liu	Tsinghua University, China
Guanfeng Liu	Macquarie University, Australia
Hao Liu	Hong Kong University of Science and Technology, China
Jinghui Liu	University of Melbourne, Australia
Kangzheng Liu	Huazhong University of Science and Technology, China
Xinpeng Liu	Dalian University of Technology, China
Yang Liu	Dalian University of Technology, China

Yue Liu	Data61, CSIRO, Australia
Sin Kit Lo	Data61, CSIRO, Australia
Emiliano Lorini	French National Centre for Scientific Research, France
Qinghua Lu	Data61, CSIRO, Australia
Dickson Lukose	Tabcorp Holdings Limited, Australia
Jieting Luo	Zhejiang University, China
Sreenivasan M.	International Institute of Information Technology, India
Chuan Ma	Zhejiang Lab, China
Hui Ma	Victoria University of Wellington, New Zealand
Pathum Chamikara Mahawaga Arachchige	Data61, CSIRO, Australia
Michael Maher	Reasoning Research Institute, Australia
Vikash Maheshwari	Universiti Teknologi PETRONAS, Malaysia
Rohana Mahmud	Universiti Malaya, Malaysia
Eric Martin	University of New South Wales, Australia
Sanparith Marukatat	National Electronics and Computer Technology Center, Thailand
Atiya Masood	Iqra University, Pakistan
Nur Ulfa Maulidevi	Bandung Institute of Technology, Indonesia
Alan Mccabe	Griffith University, Australia
Md Humaion Kabir Mehedi	BRAC University, Bangladesh
Qingxin Meng	University of Nottingham - Ningbo, China
Jian Mi	Yangzhou University, China
Lynn Miller	Monash University, Australia
Muhammad Syafiq Mohd Pozi	Universiti Utara Malaysia, Malaysia
Kristen Moore	Data61, CSIRO, Australia
Fernando Mourao	SEEK, Australia
Lailil Muflikhah	Universitas Brawijaya, Indonesia
Ganesh Neelakanta Iyer	National University of Singapore, Singapore
M. A. Hakim Newton	University of Newcastle, Australia
Phi Le Nguyen	Hanoi University of Science and Technology, Vietnam
Thanh Thi Nguyen	Deakin University, Australia
Nianwen Ning	Henan University, China
Hussain Nyeem	Military Institute of Science and Technology, Bangladesh
Kouzou Ohara	Aoyama Gakuin University, Japan
Nurul Aida Osman	Universiti Teknologi PETRONAS, Malaysia
Takanobu Otsuka	Nagoya Institute of Technology, Japan
Abiola Oyegun	Birmingham City University, UK

Maurice Pagnucco	University of New South Wales, Australia
Shirui Pan	Griffith University, Australia
Anum Paracha	Birmingham City University, UK
Anand Paul	Kyungpook National University, South Korea
Pengfei Pei	Chinese Academy of Sciences, China
Shengbing Pei	Anhui University, China
Songwen Pei	University of Shanghai for Science and Technology, China
Tao Peng	UT Southwestern Medical Center, USA
Arif Perdana	Monash University, Indonesia
Laurent Perrussel	University of Toulouse, France
Duc Nghia Pham	MIMOS Berhad, Malaysia
Ioannis Pierros	Aristotle University of Thessaloniki, Greece
Chiu Po Chan	Universiti Malaysia Sarawak, Malaysia
Thadpong Pongthawornkamol	Kasikorn Business-Technology Group, Thailand
Surya Prakash	University of the South Pacific, Fiji
Mauridhi Hery Purnomo	Institut Teknologi Sepuluh Nopember, Indonesia
Ayu Purwarianti	Bandung Institute of Technology, Indonesia
Qi Qi	Hainan University, China
Shiyou Qian	Shanghai Jiao Tong University, China
Jianglin Qiao	Western Sydney University, Australia
Chuan Qin	Baidu, China
Lyn Qiu	Shanghai Jiao Tong University, China
Joel Quinqueton	Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, France
Teeradaj Racharak	Japan Advanced Institute of Science and Technology, Japan
Jessica Rahman	CSIRO, Australia
Mohammad Shahriar Rahman	United International University, Bangladesh
Srikari Rallabandi	Vidya Jyothi Institute of Technology, India
Tian Ran	Northwest Normal University, China
Annajiat Alim Rasel	BRAC University, Bangladesh
Mahmood Rashid	Griffith University, Australia
Md Saifullah Razali	University of Wollongong, Australia
Farid Razzak	New York University, USA
Karuna Reddy	University of the South Pacific, Fiji
Fenghui Ren	University of Wollongong, Australia
Jiankang Ren	Dalian University of Technology, China
Yongli Ren	RMIT University, Australia
Yuheng Ren	Jimei University, China
Mark Reynolds	University of Western Australia, Australia
Jia Rong	Monash University, Australia

Yi Rong	Wuhan University of Technology, China
Liat Rozenberg	Griffith University, Australia
Ji Ruan	Auckland University of Technology, New Zealand
Filip Rusak	CSIRO, Australia
Arun Anand Sadanandan	SEEK Limited, Australia
Khairun Saddami	Universitas Syiah Kuala, Indonesia
Payel Sadhukhan	TCG CREST, India
Sofia Sahab	Kyoto University, Japan
Chiaki Sakama	Wakayama University, Japan
Ario Santoso	Independent, The Netherlands
Muhamad Saputra	Monash University, Indonesia
Yunita Sari	Universitas Gadjah Mada, Indonesia
Anto Satriyo Nugroho	National Research and Innovation Agency, Indonesia
Abdul Sattar	Griffith University, Australia
Thanveer Shaik	University of Southern Queensland, Australia
Lin Shang	Nanjing University, China
Nandita Sharma	Australian Government, Australia
Dazhong Shen	University of Science and Technology of China, China
Yifan Shen	University of Illinois Urbana-Champaign, USA
Chenwei Shi	Tsinghua University, China
Kaize Shi	University of Technology Sydney, Australia
Xiaolong Shi	Guangzhou University, China
Zhenwei Shi	Beihang University, China
Kazutaka Shimada	Kyushu Institute of Technology, Japan
Yanfeng Shu	CSIRO, Australia
Harvinder Singh	Torrens University, Australia
Ronal Singh	Data61, CSIRO, Australia
Patrick Chin Hooi Soh	Multimedia University, Malaysia
Chattrakul Sombattheera	Mahasarakham University, Thailand
Insu Song	James Cook University, Australia
Xin Song	Hebei University, China
Pokpong Songmuang	Thammasat University, Thailand
Lay-Ki Soon	Monash University Malaysia, Malaysia
Bela Stantic	Griffith University, Australia
Markus Stumptner	University of South Australia, Australia
Guoxin Su	University of Wollongong, Australia
Ruidan Su	Shanghai Jiao Tong University, China
Xingchi Su	Zhejiang Lab, China
Jie Sun	Nanjing Xiaozhuang University, China
Xin Sun	Zhejiang Lab, China

Ying Sun	Hong Kong University of Science and Technology, China
Yongqian Sun	Nankai University, China
Boontawee Suntisrivaraporn	DTAC, Thailand
Thepchai Supnithi	National Electronics and Computer Technology Center, Thailand
Chang Wei Tan	Monash University, Australia
David Taniar	Monash University, Australia
Thitipong Tanprasert	Assumption University of Thailand, Thailand
Xiaohui Tao	University of Southern Queensland, Australia
Sotarat Thammaboosadee	Mahidol University, Thailand
Truong Thao Nguyen	National Institute of Advanced Industrial Science and Technology, Japan
Bui Thi-Mai-Anh	Institut de la Francophonie pour l'Informatique, Vietnam
Michael Thielscher	University of New South Wales, Australia
Hung Nghiep Tran	National Institute of Informatics, Japan
Jarrod Trevathan	Griffith University, Australia
Bambang Riyanto Trilaksono	Institut Teknologi Bandung, Indonesia
Bayu Trisedya	SEEK, Australia
Eric Tsui	Hong Kong Polytechnic University, China
Shikui Tu	Shanghai Jiao Tong University, China
Ayad Turky	University of Sharjah, United Arab Emirates
Takahiro Uchiya	Nagoya Institute of Technology, Japan
Khimji Vaghjiani	Torrens University, Australia
Hans van Ditmarsch	University of Toulouse, France
Miroslav Velev	Aries Design Automation, USA
Agustinus Waluyo	La Trobe University, Australia
Biao Wang	Zhejiang Lab, China
Chao Wang	HKUST Fok Ying Tung Research Institute, China
Chen Wang	National Institute of Water and Atmospheric Research, New Zealand
Hao Wang	Monash University, Australia
Hao Wang	Nanyang Technological University, Singapore
Li Wang	Henan University, China
Shuxia Wang	Northwestern Polytechnical University, China
Weiqing Wang	Monash University, Australia
Xiangmeng Wang	University of Technology Sydney, Australia
Xinxhi Wang	Shanghai University, China
Yuxin Wang	Dalian University of Technology, China
Zhen Wang	Zhejiang Lab, China
Ian Watson	University of Auckland, New Zealand

Xian Wei	East China Normal University, China
Xiao Wei	Shanghai University, China
Manuel Weiss	SEEK, Australia
Paul Weng	UM-SJTU Joint Institute, China
Derry Wijaya	Monash University Indonesia, Indonesia
Tri Kurniawan Wijaya	Huawei Ireland Research Centre, Ireland
Arie Wahyu Wijayanto	Politeknik Statistika STIS, Indonesia
Wayne Wobcke	University of New South Wales, Australia
Daphne Wong-A-Foe	Leiden University, The Netherlands
Sartra Wongthanavasu	Khon Kaen University, Thailand
Brendon J. Woodford	University of Otago, New Zealand
Huiwen Wu	Ant Group, China
Ou Wu	Tianjin University, China
Shiqing Wu	University of Technology Sydney, Australia
Xing Wu	Shanghai University, China
Yutong Wu	CSIRO, Australia
Pierre-Henri Wuillemin	LIP6, Sorbonne University, France
Zhanhao Xiao	Guangzhou University, China
Zhuoyu Xiao	Hunan Industry Polytechnic, China
Kaibo Xie	University of Amsterdam, The Netherlands
Ming Xu	Xi'an Jiaotong-Liverpool University, China
Shuxiang Xu	University of Tasmania, Australia
Yongxiu Xu	Chinese Academy of Sciences, China
Zenghui Xu	Zhejiang Lab, China
Hui Xue	Southeast University, China
Chao Yang	University of Technology Sydney, Australia
Chunming Yang	Southwest University of Science and Technology, China
Fengyu Yang	Nanchang Hangkong University, China
Haoran Yang	University of Technology Sydney, Australia
Liu Yang	Central South University, China
Tianle Yang	Osaka University, Japan
Yi Yang	Hefei University of Technology, China
Yuan Yao	University of Nottingham - Ningbo, China
Roland Yap	National University of Singapore, Singapore
Xuefei Yin	Griffith University, Australia
Dianer Yu	University of Technology Sydney, Australia
Hang Yu	Shanghai University, China
Ting Yu	Zhejiang Lab, China
Youren Yu	Beijing Information Science and Technology University, China

Weiwei Yuan	Nanjing University of Aeronautics and Astronautics, China
Lin Yue	University of Newcastle, Australia
Evi Yulianti	Universitas Indonesia, Indonesia
Intan Nurma Yulita	Padjadjaran University, Indonesia
Nayyar Zaidi	Deakin University, Australia
Chengwei Zhang	Dalian Maritime University, China
Daokun Zhang	Monash University, Australia
Du Zhang	California State University, USA
Haibo Zhang	Kyushu University, Japan
Haijun Zhang	Harbin Institute of Technology, China
Huan Zhang	China University of Geosciences, China
Le Zhang	University of Science and Technology of China, China
Leo Zhang	Griffith University, Australia
Liying Zhang	China University of Petroleum, China
Min-Ling Zhang	Southeast University, China
Mingyue Zhang	Southwest University, China
Peng Zhang	Shandong University, China
Qi Zhang	University of Science and Technology of China, China
Shenglin Zhang	Nankai University, China
Wei Emma Zhang	University of Adelaide, Australia
Wen Zhang	Beijing University of Technology, China
Xianhui Zhang	Hangzhou Normal University, China
Xiaobo Zhang	Southwest Jiaotong University, China
Xinghua Zhang	Chinese Academy of Sciences, China
Yuhong Zhang	Hefei University of Technology, China
Yunfeng Zhang	Shandong University of Finance and Economics, China
Zili Zhang	Deakin University, Australia
Dengji Zhao	ShanghaiTech University, China
Ruilin Zhao	Huazhong University of Science and Technology, China
Yijing Zhao	Chinese Academy of Sciences, China
Jianyu Zhou	Nankai University, China
Shuigeng Zhou	Fudan University, China
Xin Zhou	Nanyang Technological University, Singapore
Yun Zhou	National University of Defense Technology, China
Enqiang Zhu	Guangzhou University, China
Guohun Zhu	University of Queensland, Australia

Jingwen Zhu	Nankai University, China
Liang Zhu	Hebei University, China
Nengjun Zhu	Shanghai University, China
Xingquan Zhu	Florida Atlantic University, USA
Yanming Zhu	Griffith University, Australia

Additional Reviewers

Angelov, Zhivko
Azam, Basim
Burgess, Mark
Cao, Xuemei
Chan, Chee-Yong
Chandra, Abel
Chen, Xiaohong
Clifton, Ava
Duan, Jiaang
Ebrahimi, Ali
Fang, Han
Fei, Wu
Fodor, Gabor Adam
Folkman, Lukas
Geng, Chuanxing
Guo, Ruoyu
Guo, Siyuan
Hammond, Lewis
Han, Xin
Hao, Chen
Haruta, Shuichiro
He, Haoyu
He, Tao
He, Zhengqi
Hu, Han Wen
Hua, Qin
Hua, Yuncheng
Huang, Renhao
Hung, Nguyen
Jiang, Zhaohui
Li, Jingyang
Li, Xiang
Liga, Davide
Lin, Songtuan
Liu, Chuan
Liu, Hongquan
Liu, Yongchang
Liu, Yutao
Liu, Zhaorui
Ma, Jiaxuan
Mataeimoghadam, Fereshteh
Mayer, Wolfgang
Mezza, Stefano
Mohamed Muzammil, Mohamed Mufassirin
Mu, Chunjiang
Nikafshan Rad, Hima
Nwe, Hlaing Myat
Pan, Chaofan
Peng, Lilan
Perera, Isuri
Rahman, Julia
Reddy, Emmenual
Ren, Siyue
Ren, Yixin
Schwenker, Friedhelm
Selway, Matt
Semenov, Ivan
Shiri, Fatemeh
Singh, Priyanka
Singh, Satyanand
Smith, Jeff
Song, Zhihao
Soni, Bhanu Pratap
Tan, Hongwei
Tang, Jiaqi
Viriyavisuthisakul, Supatta
Wang, Luzhi
Wang, Mengyan
Wang, Xiaodan

Wang, Yunyun
Wei, Tianpeng
Wu, Lingi
Wu, Shixin
Xia, Boming
Xu, Dalai
Xu, Rongxin
Xu, Weilai
Yang, Yikun
Yao, Naimeng
Yin, Yifan
Yuan, Zixuan
Zaman, Rianon
Zhang, Denghui
Zhang, Junyu
Zhang, Lin
Zhang, Yunfei
Zhang, Zhenxing
Zhao, Zijun
Zheng, Xin
Zheng, Yizhen
Zhou, Zheng

Contents – Part II

Machine Learning/Deep Learning

Natural Language Processing

Optimization

Responsible AI/Explainable AI

Machine Learning/Deep Learning

A Spatial Interpolation Method Based on BP Neural Network with Bellman Equation

Liang Zhu, Haiyang Wei(✉), Xin Song, Yonggang Wei, and Yu Wang(✉)

Hebei University, Baoding 071002, Hebei, China
hywhbu@163.com, wy@hbu.edu.cn

Abstract. Spatial interpolation is a valuable technique that uses the data of a sample set to estimate the property values at unsampled locations. Neural networks for spatial interpolation can capture spatial trends effectively; however, they may not be optimal when a strong local correlation is present, which leads to unreliable outcomes. Neural Network Residual Kriging methods use Kriging to handle residuals, assuming strict conditions such as the stationarity of the random field and stable spatial variability. In many applications without these strict conditions, however, those Neural Network interpolation methods have limitations for obtaining highly accurate estimates. To address this problem, in this paper, we propose a new spatial interpolation method, called NNRB, based on the mechanisms of BP Neural Network with Bellman Equation. Firstly, our NNRB method employs a BP neural network for capturing nonlinear relationships and spatial trends in the data of a sample set. Secondly, NNRB uses Bellman Equation to handle residuals by accounting for interactions between multiple adjacent data and reducing the influence of distant data on the current data. Our NNRB method is utilized for a system of soil testing and formulated fertilization for intelligent agriculture. We compared NNRB with four state-of-the-art interpolation methods, and the results show that our NNRB method outperforms the three methods significantly and is highly competitive with one approach.

Keywords: Spatial Interpolation · Backpropagation Neural Network · Bellman Equation · Interpolation Residual · Data Gridding · Markov Reward Process

1 Introduction

Continuous spatial data (e.g., geological data, meteorological data, etc.) is a fundamental requirement for various projects, systems or scientific studies. In many applications (say, formulating fertilizer recommendations), however, high-density sampling cannot be performed due to time and capital costs, technical means, terrain factors, etc. Geographical First Law suggests that characteristic values of spatially close points are more likely to be similar, while distant points are less likely to have similar values [1]. Based on this Law, various spatial interpolation methods have been proposed to fill gaps in incomplete data and applied in hydrology [2], ecology [3], agriculture [4], economics [5], and other fields [6]. For instance, one of the most popular methods for spatial modeling and prediction is Ordinary Kriging (OK) [7], which is a regression algorithm using

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 3–15, 2024.
https://doi.org/10.1007/978-981-99-7022-3_1

the covariance function to estimate a random process/field [8]. Two conditions must be met for Kriging method to be applicable: Firstly, the random field has a mathematical expectation that is location-invariant. Secondly, a suitable covariance function is found to describe the spatial correlation structure of the random field.

A three-layer neural network can theoretically approximate any complex function [9]; thus, several kinds of Neural Network methods are studied by using the non-linear feedback mechanism to interpolate the estimated value of a predicted property (e.g., nitrogen, phosphorus, or potassium, etc.). For example, Backpropagation Neural Network Interpolation (BPNNI) [10] and Neural Network Residual Kriging (NNRK) [11] are two state-of-the-art ones in the family of Neural Network methods. However, the interpolated result for a certain point in BPNNI-like methods is exclusively derived from auxiliary data at that point, regardless of the spatial autocorrelation of the surrounding measurement data [12]. The NNRK-like ones entail forecasting trends using a neural network, computing predicted residuals, and subsequently fitting these residuals using OK; however, OK has its own limitations, such as strict assumptions as mentioned above and difficulty in finding a reliable geostatistical model that suits all residual data.

To address the above problems, we discuss a new kind of spatial interpolation method. The contributions of this paper are summarized below: (1) For a spatial sample dataset with the values of locations and properties, we propose a novel spatial interpolation method by employing the mechanisms of BP neural network and Bellman Equation, namely Neural Network Residual Bellman (NNRB), which can be used to overcome the limitations of some neural network interpolation methods (say, NNRK, and BPNNI). (2) We utilize the gridding method to discretize the prediction errors (i.e., residuals) between the predicted values of a BP neural network model and the actual observed/measured values in the sample dataset, and then we handle the residuals by updating the grid iteratively with the Bellman Eq. (3) Our NNRB method is applied to a system of soil testing and formulated fertilization for intelligent agriculture; moreover, extensive experiments are conducted to compare NNRB with four methods over four datasets of soil samples.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. In Sect. 3, our method NNRB is proposed. Section 4 presents the experimental results. Finally, Sect. 5 concludes the paper.

2 Related Work

In this section, we provide a brief review of related work concerning spatial datasets. In 2019, Sergeev et al. used an artificial neural network (ANN) to simulate the non-linear trends of extensive data and then modeled the residuals using Ordinary Kriging [13]. Their model showed higher accuracy in estimation compared to traditional BPNNI models. In 2020, Zhu et al. utilized conditional generative adversarial neural networks (cGANs) for altitude interpolation in China, and the experimental results in [14] showed that cGANs have excellent performance in the fields of temperature, rainfall, altitude and so on. In 2023, Luo developed a Generalized Heterogeneity Model (GHM) to enhance the interpolation accuracy of marine chlorophyll [15]. It is also pointed out that GHM has the potential to be integrated with machine learning and advanced algorithms to improve spatial prediction accuracy in broader fields. In 2023, Lee observed challenges in applying traditional regionalization techniques to address extreme weather data acquired from unevenly distributed meteorological stations [16]. The Markov chain random field technique was employed to complete the dataset, followed by the application of the Kriging method to assess precipitation extremes. Results highlight the method's effectiveness in addressing data gaps within unmeasured regions.

Neural networks are able to fit function relationships well; however, there are significant flaws in their ability to estimate the residuals, which limits the use of neural networks for spatial interpolation purposes. We will propose a new method, namely NNRB, to make up for the flaws and improve the interpolation accuracy.

3 NNRB Method

The main idea of the Neural Network Residual Bellman (NNRB) method proposed in this paper is as follows: Firstly, the spatial *trends* $t^*_{\mathrm{BPNN}}(\cdot)$ are modeled by a BP neural network (BPNN). Secondly, the *residual* $r^*_{\mathrm{BE}}(\cdot)$ is estimated by the Bellman Equation. Finally, for a vector $\boldsymbol{u} = (c_1, c_2, \cdots, c_q)$, the NNRB *estimator* denoted by $e^*_{\mathrm{NNRB}}(\boldsymbol{u})$ is defined by Formula (1), which is the summation of the spatial trends $t^*_{\mathrm{BPNN}}(\boldsymbol{u})$ and the residual $r^*_{\mathrm{BE}}(\boldsymbol{u})$.

$$e^*_{\mathrm{NNRB}}(\boldsymbol{u}) = t^*_{\mathrm{BPNN}}(\boldsymbol{u}) + r^*_{\mathrm{BE}}(\boldsymbol{u}) \tag{1}$$

The flow diagram of the NNRB method for spatial prediction of soil nutrient data is presented in Fig. 1.

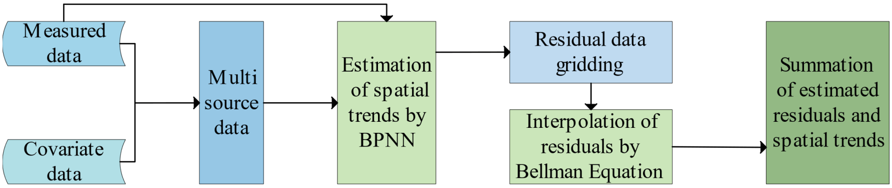

Measured data
Covariate data
Multi source data
Estimation of spatial trends by BPNN
Residual data gridding
Interpolation of residuals by Bellman Equation
Summation of estimated residuals and spatial trends

Fig. 1. Flow diagram of NNRB method for spatial prediction of soil nutrient data

3.1 BP Neural Network

Interpolation methods typically deal with small datasets. However, utilizing complex neural networks on these datasets may lead to overfitting. Despite its simplicity, the BP neural network has a strong theoretical foundation and learning mechanisms, which make it a commonly used model in interpolation problems. In the spatial interpolation problem, the sample dataset can be defined as a tuple set $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_n\}$ [15] ($|\boldsymbol{T}|$ means the number of tuples in $\boldsymbol{T}$). The schema of $\boldsymbol{T}$ is $\boldsymbol{T}(A_1, A_2, \cdots, A_d, B_1, B_2, \cdots, B_p, C_1, C_2, \cdots, C_q) = \boldsymbol{T}(\boldsymbol{S}, \boldsymbol{Z}, \boldsymbol{U})$, where $\boldsymbol{S} = (A_1, A_2, \cdots, A_d)$ is a set of independent variables, $\boldsymbol{Z} = (B_1, B_2, \cdots, B_p)$ is a set of covariates, and $\boldsymbol{U} = (C_1, C_2, \cdots, C_q)$ is a set of dependent variables. Each tuple $\boldsymbol{t}_i$ in $\boldsymbol{T}$ is defined as $\boldsymbol{t}_i = (a_{i1}, a_{i2}, \cdots, a_{id}, b_{i1}, b_{i2}, \cdots, b_{ip}, c_{i1}, c_{i2}, \cdots, c_{iq}) = (\boldsymbol{s}_i, \boldsymbol{z}_i, \boldsymbol{u}_i) \in \boldsymbol{T}$, where $\boldsymbol{s}_i = (a_{i1}, a_{i2}, \cdots, a_{id})$ is a vector of the values of independent variables, $z_i = (b_{i1}, b_{i2}, \cdots, b_{ip})$ is a vector of the covariate values, and $\boldsymbol{u}_i = (c_{i1}, c_{i2}, \cdots, c_{iq})$ is a vector of the values of the dependent variables. Therefore, $\boldsymbol{T} = \{(\boldsymbol{s}_1, z_1, \boldsymbol{u}_1), (\boldsymbol{s}_2, z_2, \boldsymbol{u}_2), \cdots, (\boldsymbol{s}_n, z_n, \boldsymbol{u}_n)\}$.

For instance, $\boldsymbol{T}$ is a soil sample set in our experiment, in which the d ($= 2$) independent variables are location variables such as the *latitude* and *longitude*, and the covariates are various terrain factors such as *elevation*, *slope*, *aspect*, and *slope gradient change rate*, while the dependent variables are the properties such as *organic matter*, *total nitrogen*, *available phosphorus*, *quick-acting potassium*, and *slow-acting potassium.* For a interpolation point $\boldsymbol{t} = (a_1, a_2, b_1, b_2, \cdots, b_p, x_1, x_2, \cdots, x_q) = (\boldsymbol{s}, z, \boldsymbol{x})$ where $\boldsymbol{s} = (a_1, a_2)$ and $z = (b_1, b_2, \cdots, b_p)$ are known, we will predict $\boldsymbol{x} = (x_1, x_2, \cdots, x_q)$ by the sample set $\boldsymbol{T}$ with our NNRB method.

We design the structure of the BP neural network (BPNN) used in the NNRB method as follows. The number of input layer nodes is determined by the dimensionality of the longitude and latitude and the dimensionality of the geographic attributes. The number of nodes in the hidden layer in our model is defined by the empirical Formula (2).

$$H = (I + O)^{1/2} + \alpha \tag{2}$$

where H is the number of hidden layer nodes, I is the number of input layer nodes, O is the number of output layer nodes, and α is an integer [17]. We predict one property at a time, then $O = 1$, i.e., the output layer contains one node. To eliminate the influence of the value units on the NNRB method, the data need to be normalized by

$$w_i^* = (w_i - w_{min})/(w_{max} - w_{min}) \tag{3}$$

where w_{max} and w_{min} are the maximum and minimum values of each dimension value w_i (e.g., w_i may be the value of latitude or nitrogen) for a sample point.

After the dataset $\boldsymbol{T}$ is cleaned, normalized and randomly shuffled, $\boldsymbol{T}$ will be divided into five subsets $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, $\boldsymbol{T}_3$, $\boldsymbol{T}_4$ and $\boldsymbol{T}_{5,}$ satisfying $\boldsymbol{T} = \boldsymbol{T}_1 \cup \cdots \cup \boldsymbol{T}_5$, $\boldsymbol{T}_i \cap \boldsymbol{T}_j = \emptyset$ and $|\boldsymbol{T}_i| = |\boldsymbol{T}_j|$ ($1 \leq i \neq j \leq 5$). Then we obtain four datasets: $\boldsymbol{D}_1 = \boldsymbol{T}_2 \cup \boldsymbol{T}_3 \cup \boldsymbol{T}_4$, $\boldsymbol{D}_2 = \boldsymbol{T}_1 \cup \boldsymbol{T}_3 \cup \boldsymbol{T}_4$, $\boldsymbol{D}_3 = \boldsymbol{T}_1 \cup \boldsymbol{T}_2 \cup \boldsymbol{T}_4$ and $\boldsymbol{D}_4 = \boldsymbol{T}_1 \cup \boldsymbol{T}_2 \cup \boldsymbol{T}_3$. For each i ($1 \leq i \leq 4$), let $\boldsymbol{D}_i$ be the training set, while both $\boldsymbol{T}_i$ and $\boldsymbol{T}_5$ be the test set. We use $\boldsymbol{D}_i$ to train a BP neural network, obtain predicted values $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{ik})$ for the subset $\boldsymbol{T}_i$ and the prediction $\boldsymbol{y}'_i = (y'_{i1}, \cdots, y'_{ik})$ for $\boldsymbol{T}_5$. Then, we calculate the residuals $\{\boldsymbol{\varepsilon}_i$: $i = 1, \cdots, 4\}$ for the four subsets $\{\boldsymbol{T}_i$: $i = 1, \cdots, 4\}$ using their actual observed/measured values $\boldsymbol{v}_i = (v_{i1}, \cdots, v_{ik})$, that is, $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \cdots, \varepsilon_{ik}) = \boldsymbol{y}_i - \boldsymbol{v}_i = (y_{i1} - v_{i1}, \cdots, y_{ik} - v_{ik})$. Let residual $\boldsymbol{\varepsilon}_5$ for subset $\boldsymbol{T}_5$ be an empty set (i.e., $\boldsymbol{\varepsilon}_5 = \emptyset$). Thus, we obtain the five residual datasets with the format $(\boldsymbol{S}, \boldsymbol{\varepsilon})$, in which $\boldsymbol{S} = (\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_d)$ is a set of d location coordinates (say, $d = 2$ for the *latitude* and *longitude* in our experiments) of tuples, and $\boldsymbol{\varepsilon}$ is the corresponding residual values for the tuples.

3.2 Data Gridding

As depicted in Fig. 2, for the sample set $\boldsymbol{T}$, a grid technique will be used in our NNRB method to handle the residuals [18]. Firstly, the minimum Euclidean distance $d_{min} = min\{d(\boldsymbol{t}_i[\boldsymbol{S}], \boldsymbol{t}_j[\boldsymbol{S}])$: $1 \leq i \neq j \leq |\boldsymbol{T}|\}$ is applied as the diagonal length of the grid cell, where $d(\boldsymbol{t}_i[\boldsymbol{S}], \boldsymbol{t}_j[\boldsymbol{S}])$ is the Euclidean distance between the location coordinates (e,g., *latitude* and *longitude*) in the residual dataset with the format ($\boldsymbol{S}$, $\boldsymbol{\varepsilon}$). Secondly, according to the location coordinates of each residual value, its relative position was identified within the grid. For arbitrary t_i and t_j in a cell, therefore, if the diagonal length of this cell is less than or equal to d_{min}, then t_i and t_j represent the same point. Finally, each grid cell is set to contain at most one residual value, which in general leads to a sparse grid.

In Fig. 2(a), gray and blue colors are employed to distinguish empty and non-empty cells, respectively. As an efficiency enhancement strategy, the diagonal length of the cell is increased progressively by a certain factor; meanwhile, the placement records of residual values are maintained. In case of a location conflict, subsequent points are placed in an adjacent position to the conflicting point, such as its top, bottom, left or right side. We ensure the overall relative position of each point remains unchanged but limit the number of moved points to within 2% of the total points. The maximum diagonal length of the cells within the range is used in the scaled grid, as demonstrated in Fig. 2(b). We designate the above subset $\boldsymbol{T}_5$ of $\boldsymbol{T}$ as the test set and use an orange cell to illustrate the location of a point from $\boldsymbol{T}_5$ in Fig. 2(c). Under the condition of keeping the original grid data value unchanged, the gray grid cells are iteratively updated to describe the updated situation. The final result of the iteration is represented using a green cell, as shown in Fig. 2(d), where the interpolation outcome of a point from $\boldsymbol{T}_5$ is displayed in the pink cell.

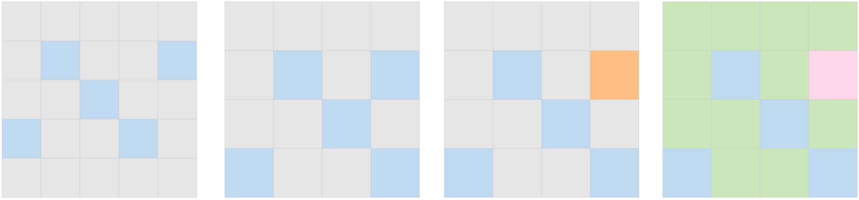

(a) placement (b) scaling (c) partitioning (d) completed

Fig. 2. Schematic of data gridding

3.3 Bellman Equation Processes Residuals

Considering a stochastic/random process, if the conditional probability distribution of the future state given the present and past states depends only on the present state and is independent of the past states of the process, then this process is called a Markov process [19], which can be defined by Formula (4).

$$P(\omega_{t+1}|\omega_t) = P(\omega_{t+1}|\omega_1, \cdots, \omega_t) \tag{4}$$

Motivated by the idea of the Markov process, we assume that residual values are influenced only by the values of their neighbors [20]. A cell with adjacent edges is defined as a neighboring cell in this paper. Figure 3(a) shows the spatial relationship between the sampling points and their neighbors. Light-colored cells represent the neighbors of dark-colored cells. A Markov reward process (MRP) denoted by ($\mathit{\Omega}$, $\mathcal{P}$, $\mathit{\Phi}$, γ) is a valuable extension of Markov process, where Ω is the finite set of states. The state transition matrix, denoted as $\mathcal{P}$, describes the conditional probability that the state ω will transition to another state $\omega^{'}$ at time $t + 1$, given its current state at time t [21], which is expressed by Formula (5).

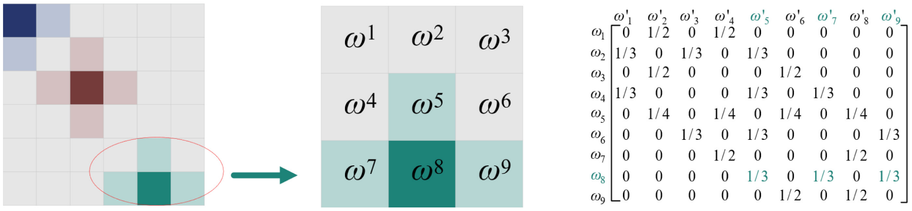
$$\mathcal{P}_{\omega\omega'} = P(\omega_{t+1} = \omega' | \omega_t = \omega) \tag{5}$$

In the residual grid, as shown in Fig. 3, each cell can be viewed as a state, and the state transition probability is used to measure the proportion of influence that neighboring cells have on the current cell [22]. For simplicity, we assume that each neighbor has the same influence. For example, if cell ω_8 has three neighbors, $\omega^{'}_5$, $\omega^{'}_7$, and $\omega^{'}_9$, then the influence of $\omega^{'}_5$, $\omega^{'}_7$, and $\omega^{'}_9$ on ω_8 is equal, with a proportion of 1/3 for each, as shown in Fig. 3(b) and Fig. 3(c).

In $(\Omega, \mathcal{P}, \Phi, \gamma)$, Φ is defined by Formula (6), which is the expected reward that state ω_t at time t will receive at the next time step $(t + 1)$, where R_{t+1} is the actual reward that the state ω_t at time t will receive at the next time step $(t + 1)$.

$$\Phi(\omega) = E(R_{t+1} | \omega_t = \omega) \tag{6}$$

In $(\Omega, \mathcal{P}, \Phi, \gamma)$, γ $(0 \leq \gamma \leq 1)$ is the discount factor. We use γ to measure the degree of spatial correlation contained within the residual grid [19].



$$\begin{array}{c|ccccccccc} & \omega'_1 & \omega'_2 & \omega'_3 & \omega'_4 & \omega'_5 & \omega'_6 & \omega'_7 & \omega'_8 & \omega'_9 \\ \hline \omega_1 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ \omega_2 & 1/3 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ \omega_3 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ \omega_4 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ \omega_5 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ \omega_6 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ \omega_7 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ \omega_8 & 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ \omega_9 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \end{array}$$

(a) neighborship (b) equal-weight (c) weight matrix

Fig. 3. Illustration of neighboring cells and their influence

In a Markov reward process, the return G_t is defined as the sum of rewards discounted by a factor γ, which is obtained from a starting state ω_t at time t to the terminal state ω_{t+K+1}, as shown in Formula (7).

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^K R_{t+K+1} = \sum\nolimits_{k=0}^{K} \gamma^k R_{t+k+1} \tag{7}$$

The expected return of a state is called its value function, defined by Formula (8).

$$v(\omega) = E(G_t | \omega_t = \omega) \tag{8}$$

In our residual grid, we use the value function of a state as the value of the grid, which is calculated by using the Bellman Equation with Formula (9).

$$v(\omega) = \Phi(\omega) + \gamma \sum\nolimits_{\omega' \in \Omega} P_{\omega\omega'} v(\omega') \tag{9}$$

The discount factor γ is utilized to quantify the correlation among points. We traverse the scaled interpolation grid and remove the value of a non-empty cell with a probability based on the subsampling ratio. This process results in the generation of a subsampled grid after a single pass of traversal. We determine the best value of γ on the subsampled grid utilizing a random search method and use it as the optimal discount factor for the entire grid.

The value of the current cell is calculated as the weighted average of its neighbors. For simplicity, the values of a reward function $\Phi(\omega)$ for all cells are set to 0 during the calculation of all cell values. As an example, Formula (10) is the result of substituting actual values into Formula (9) for cell ω_8 in Fig. 3(b).

$$v(\omega_8) = \gamma(\frac{1}{3}v(\omega_5') + \frac{1}{3}v(\omega_7') + \frac{1}{3}v(\omega_9')) \tag{10}$$

Formula (9) is used to update the subsampled grid. For a grid with $I \times J$ cells, we use δ to record the magnitude of each update. In Formula (11), $v(i, j)$ and $v^*(i, j)$ represent the values before and after the update at position (i, j). When the magnitude is no more than a threshold, we consider the model to have converged and stopped iterating; then the interpolated residual grid is returned.

$$\delta = \frac{1}{I \times J}\sum\nolimits_{i=1}^{I}\sum\nolimits_{j=1}^{J}\left|v(i,j) - v^*(i,j)\right| \tag{11}$$

The best value of γ is obtained through the random search function that minimizes the root mean square error of the subsampled grid interpolation. In the entire residual grid, we use this γ, Formula (9) and Formula (11) to update the value of each cell iteratively.

```
Algorithm BE             // Bellman-Equation updates grid values
Input $N$, G, $\gamma$, $\tau$      // $N$ is Max iterations, G is input grid, $\gamma$ is the discount factor,
                         // $\tau$ is convergence threshold
Output InterpolatedGrid  // output grid
3-dim array NeighborsGrid;        // store the values of neighboring cells
2-dim array CountNeighbors;       // store the number of neighbors
2-dim array IntermediateGrid;     // intermediate grid for interpolation
2-dim array WeightArray;          // as weight vector for each neighboring
For $i = 1$ To $N$                    // iterative calculation
    update NeighborsGrid;         // compute neighboring cell values and weights
    calculate the inner product;  // for the NeighborsGrid and WeightArray
    calculate multiplication;     // for the inner product and CountNeighbors
    check for difference $\delta$;      // check the magnitude using Formula (12)
    If $\delta \le \tau$                  // no more than the convergence threshold
        stop the update;
    Else
        update IntermediateGrid with the new grid;
    End If
End For
Return InterpolatedGrid;          // returning the interpolation result.
```

4 Experimental Results

The program is written using Python 3.11 and PyTorch 1.13.1 CPU version, and the experiments are conducted on a PC with an Intel(R) Core(TM) i5-10505 CPU @ 3.20 GHz and 16 GB of RAM.

4.1 Parameters and Settings

Four real-world soil datasets from different regions of China are used in our experiments, and the properties of the four datasets include *organic matter* (or *om* for short), *total nitrogen* (*tn*), *available phosphorus* (*ap*), *quick-acting potassium* (*qak*), or *slow-acting potassium* (*sak*). The dataset Tangshan (*om*, *tn*, *ap*, *qak*, *sak*) comes from [23]. The three datasets Togtoh (*tn*, *ap*, *qak*), Hohhot (*tn*, *ap*, *qak*), and Baiyin (*tn*, *ap*, *qak*) can be obtained from [24], and the values were extracted using ArcGIS software.

For each dataset, we conducted cross-validation by randomly selecting 80% of the tuples from it for spatial interpolation and using the remaining 20% for validation. For the same dataset and property, we ensured consistent use of the same training and testing sets. The Root Mean Square Error (*RMSE*) defined by Formula (12) are used to measure the accuracy of each interpolation method, where n is the number of sample points involved in the validation, $x^{'}_{i}$ means the predicted value of the i-th predicted point, and x_i is the actual observed/measured value of the i-th predicted point. A smaller *RMSE* indicates a higher accuracy of the measurement data.

$$RMSE = \left((1/n)\sum\nolimits_{i=1}^{n}(x_i - x_i^{\prime})^2)^{1/2}\right) \tag{12}$$

The parameters used in our method (or others) over four soil datasets include data size, number of training epochs, and maximum expanded ratio of initial residual grid. The parts of these parameters are illustrated in Table 1. *Moran's I* and the optimal value of γ for the residual grid of each property in different datasets are illustrated in Table 2.

Table 1. Parameters of BP Neural Network and Data Gridding

	Tangshan	Togtoh	Baiyin	Sanyuan
Data size	450	2000	1740	670
Epochs	200	450	350	200
Expanded ratio	726	349	519	284

We will compare our NNRB method with four representative or state-of-the-art interpolation methods: OK [7], CAIDWR [23], BPNNI [10], and NNRK [11]. The performance of the baseline methods was evaluated on various datasets and properties, and we determined their optimal parameters. Subsequently, the best results of the baseline methods were compared with our proposed approach. Specifically, we computed the residual prediction value $\boldsymbol{\varepsilon}_5$ and added it to $\boldsymbol{y}^{'}_{i}$ $(1 \leq i \leq 4)$. This step produced four sets of NNRK results, which were then averaged to obtain the final prediction outcome.

Table 2. Parameters of *Moran's I* and γ used in Bellman Equation

		om	*tn*	*ap*	*qak*	*sak*
Tangshan	*Moran's I*	0.3016	0.4988	0.4077	0.1172	0.5314
	γ	0.5418	0.4926	0.4127	0.3633	0.2998
Togtoh	*Moran's I*	-	0.8659	0.7720	0.7886	-
	γ	-	0.2797	0.3314	0.1091	-
Baiyin	*Moran's I*	-	0.8463	0.7266	0.7033	-
	γ	-	0.1942	0.3655	0.3716	-
Sanyuan	*Moran's I*	-	0.5794	0.8255	0.8651	-
	γ	-	0.4122	0.4126	0.4371	-

4.2 Results of Spatial Interpolation Over Soil Nutrient Datasets

Table 3 illustrates the experiment results with five spatial interpolation methods over the four datasets. Our NNRB method outperforms the other four methods over the Tangshan dataset with five properties and the Baiyin dataset with three properties. For Togtoh with three properties and Sanyuan with three properties, our NNRB method outperforms the OK, NNI, and NNRK methods; meanwhile, our NNRB method and CAIDWR have similar performances, in which the CAIDWR method slightly outperforms our NNRB for Togtoh (*tn*, *ap*) and Sanyuan (*tn*), but slightly underperforms our NNRB for other properties over these two datasets. Moreover, the accuracy of our NNRB is much better than that of the five OK-like and IDW-like methods as the baselines of the CAIDWR method in [23]; the results are omitted due to space limitations.

The CAIDWR method clusters sample points with similar property values and uses only relevant samples in the cluster to minimize the impact of irrelevant samples on the interpolation results. It optimizes alpha parameters based on local spatial patterns and adapts results using data trends. These factors actually make the interpolation of CAIDWR higher accuracy than five state-of-the-art OK-like and IDW-like methods. The NNRK overcomes the shortcomings of the BPNNI by using OK to further fit the residuals; compared to BPNNI, NNRK has higher interpolation accuracy; meanwhile, the results in Table 3 also confirm the founding by Kazuya Ishitsuka [25].

The advantages of Bellman Equation used in our NNRB to handle the residual data can be seen as follows. Firstly, the recursive formulation used in Bellman Equation is effective in decomposing a complex issue into smaller ones, enhancing computational efficiency significantly. Secondly, Bellman Equation offers a framework of residual prediction that accounts for interactions between multiple adjacent data while reducing the influence of distant data on the current data, and this framework is especially relevant for soil type data, because soil properties are more spatially susceptible to anthropogenic factors than most of the continuous properties such as temperature, humidity and air pressure. Our calculation of *Moran's I* also corroborates this discovery [26]. Finally, Bellman Equation can be adjusted to different data types and models without strict requirements, making it a versatile solution for processing residual data.

Table 3. *RMSE* of five spatial interpolation methods

datasets	methods	*om*	*tn*	*ap*	*qak*	*sak*
Tangshan	OK	0.1272	0.2096	0.1406	0.1647	0.1417
	CAIDWR	0.0825	0.1360	0.0946	0.1497	0.1136
	BPNNI	0.2573	0.2605	0.2760	0.2561	0.2482
	NNRK	0.2377	0.2053	0.2272	0.2077	0.2600
	NNRB	**0.0784**	**0.0909**	**0.0813**	**0.1045**	**0.0966**
Togtoh	OK	-	0.1305	0.1127	0.0899	-
	CAIDWR	-	**0.0985**	**0.0731**	0.0863	-
	BPNNI	-	0.2572	0.2271	0.2771	-
	NNRK	-	0.1908	0.2161	0.2128	-
	NNRB	-	0.1045	0.0746	**0.0704**	-
Baiyin	OK	-	0.0810	0.1390	0.1175	-
	CAIDWR	-	0.0712	0.1032	0.0794	-
	BPNNI	-	0.2180	0.2257	0.2798	-
	NNRK	-	0.1983	0.2119	0.2229	-
	NNRB	-	**0.0699**	**0.0958**	**0.0761**	-
Sanyuan	OK	-	0.0734	0.1774	0.1247	-
	CAIDWR	-	**0.0637**	0.1294	0.1085	-
	BPNNI	-	0.2707	0.2625	0.2455	-
	NNRK	-	0.2367	0.2411	0.3210	-
	NNRB	-	0.0784	**0.1078**	**0.0945**	-

As an example, we only give the distribution map of sample points in Tangshan dataset as shown in Fig. 4 and the visualization of interpolation results on quick-acting potassium property for Tangshan dataset as shown in Fig. 5.

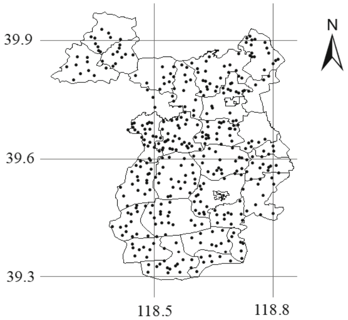


Fig. 4. Distribution map of soil nutrient sample points

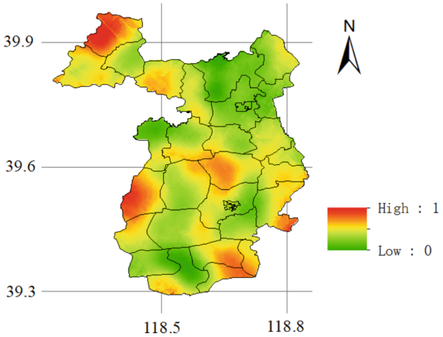


Fig. 5. Visualization of interpolation results of NNRB method

5 Conclusion

To improve the performances of the generic Neural Network interpolations and Neural Network Residual Kriging approaches, in this paper, we proposed the NNRB method based on the mechanisms of the BP Neural Network and Bellman Equation. This NNRB method employs a BP neural network to capture nonlinear relationships and spatial trends in the data of a sample set [27]; meanwhile, it uses Bellman Equation to handle residuals. NNRB method is applied in a system of soil testing and formulated fertilization for intelligent agriculture, which can be used to address complex non-linear spatial variability, the nonstationarity of the random field, and local correlation issues, and provide a high degree of accuracy [28]. We compared NNRB with four state-of-the-art interpolation methods over four datasets, and the results show that our NNRB method outperforms OK, BPNNI and NNRK significantly and it is highly competitive with the CAIDWR approach.

References

1. Tobler, W.R.: A computer movie simulating urban growth in the Detroit region. Econ. Geogr. **46**, 234–240 (1970)
2. Nurhadiyatna, A., Sunaryani, A., Sudriani, Y., Latifah, A.: 2D spatial interpolation for water quality parameter distribution in Maninjau Lake. In: 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 215–220. IEEE (2016)
3. Dai, F., Zhou, Q., Lv, Z., Wang, X., Liu, G.: Spatial prediction of soil organic matter content integrating artificial neural network and ordinary kriging in Tibetan Plateau. Ecol. Ind. **45**, 184–194 (2014)
4. Tziachris, P., Metaxa, E., Papadopoulos, F., Papadopoulou, M.: Spatial modelling and prediction assessment of soil iron using kriging interpolation with pH as auxiliary information. ISPRS Int. J. Geo Inf. **6**, 283 (2017)
5. Viana, D., Barbosa, L.: Attention-based spatial interpolation for house price prediction. In: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, pp. 540–549 (2021)
6. Tang, Y., et al.: Spatial estimation of regional PM2.5 concentrations with GWR models using PCA and RBF interpolation optimization. Remote Sens. **14**, 5626 (2022)

7. Soto, F., Navarro, F., Díaz, G., Emery, X., Parviainen, A., Egaña, Á.: Transitive kriging for modeling tailings deposits: a case study in southwest Finland. J. Clean. Prod. **374**, 133857 (2022)
8. Le, N.D., Zidek, J.V.: Statistical analysis of environmental space-time processes. Springer, New York (2006). https://doi.org/10.1007/0-387-35429-8
9. Hecht-Nielsen, R.: Kolmogorov's mapping neural network existence theorem. In: Proceedings of the International Conference on Neural Networks, pp. 11–14. IEEE Press New York, NY, USA (1987)
10. Lai, Y., et al.: Reconstructing the data gap between GRACE and GRACE follow-on at the basin scale using artificial neural network. Sci. Total. Environ. **823**, 153770 (2022)
11. Shahriari, M., Delbari, M., Afrasiab, P., Pahlavan-Rad, M.R.: Predicting regional spatial distribution of soil texture in floodplains using remote sensing data: a case of southeastern Iran. CATENA **182**, 104149 (2019)
12. Li, J., Heap, A.D.: Spatial interpolation methods applied in the environmental sciences: a review. Environ Model Softw. **53**, 173–189 (2014)
13. Sergeev, A., Buevich, A., Baglaeva, E., Shichkin, A.J.C.: Combining spatial autocorrelation with machine learning increases prediction accuracy of soil heavy metals. CATENA **174**, 425–435 (2019)
14. Zhu, D., Cheng, X., Zhang, F., Yao, X., Gao, Y., Liu, Y.: Spatial interpolation using conditional generative adversarial neural networks. Res. Output Contrib. J. **34**, 735–758 (2020)
15. Luo, P., Song, Y., Zhu, D., Cheng, J., Meng, L.: A generalized heterogeneity model for spatial interpolation. Int. J. Geograph. Inf. Sci. **37**, 634–659 (2023)
16. Lee, M.-H., Chen, Y.J.: Markov chain random field kriging for estimating extreme precipitation at unevenly distributed sites. J. Hydrol. **616**, 128591 (2023)
17. Park, H.I., Lee, S.R.: Evaluation of the compression index of soils using an artificial neural network. Comput. Geotech. **38**, 472–481 (2011)
18. Xavier, A.C., Scanlon, B.R., King, C.W., Alves, A.I.: New improved Brazilian daily weather gridded data (1961–2020). Int. J. Climatol. **42**, 8390–8404 (2022)
19. Cui, Z., Lin, L., Pu, Z., Wang, Y.: Graph Markov network for traffic forecasting with missing data. Transp. Res. Part C Emerg. Technol. **117**, 102671 (2020)
20. Vedadi, F., Shirani, S.: A map-based image interpolation method via viterbi decoding of Markov chains of interpolation functions. IEEE Trans. Image Process. **23**, 424–438 (2013)
21. Trombini, M., Solarna, D., Moser, G., Dellepiane, S.: A goal-driven unsupervised image segmentation method combining graph-based processing and Markov random fields. Pattern Recogn. **134**, 109082 (2023)
22. Colonnese, S., Rinauro, S., Scarano, G.: Bayesian image interpolation using Markov random fields driven by visually relevant image features. Sig. Process. Image Commun. **28**, 967–983 (2013)
23. Zhu, L., Hou, G., Song, X., Wei, Y., Wang, Y.: A spatial interpolation using clustering adaptive inverse distance weighting algorithm with linear regression. In: Memmi, G., Yang, B., Kong, L., Zhang, T., Qiu, M. (eds.) 15th International Conference on Knowledge Science, Engineering and Management, KSEM 2022. LNCS, Singapore, 6–8 August 2022, Proceedings, Part II, pp. 261–272. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10986-7_21
24. Liu, F., et al.: Mapping high resolution National Soil Information Grids of China. Sci. Bull. **67**(3), 328–340 (2022). https://doi.org/10.1016/j.scib.2021.10.013
25. Ishitsuka, K., Mogi, T., Sugano, K., Yamaya, Y., Uchida, T., Kajiwara, T.: Resistivity-based temperature estimation of the Kakkonda Geothermal Field, Japan, using a neural network and neural kriging. IEEE Geosci. Remote Sens. Lett. **15**, 1154–1158 (2018)
26. Zhang, C., Luo, L., Xu, W., Ledwith, V.: Use of local Moran's I and GIS to identify pollution hotspots of Pb in urban soils of Galway, Ireland. Sci. Total. Environ. **398**, 212–221 (2008)

27. Peli, R., Menafoglio, A., Cervino, M., Dovera, L., Secchi, P.: Physics-based Residual Kriging for dynamically evolving functional random fields. Stoch. Env. Res. Risk Assess. **36**, 3063–3080 (2022)
28. Agyeman, P.C., Kingsley, J., Kebonye, N.M., Khosravi, V., Borůvka, L., Vašát, R.: Prediction of the concentration of antimony in agricultural soil using data fusion, terrain attributes combined with regression kriging. Environ. Pollut. **316**, 120697 (2023)

Attention Auxiliary Supervision for Continuous Sign Language Recognition

Xiaofei Qin[1], Junyang Kong[1], Changxiang He[1], Xuedian Zhang[1](✉), Chong Ghee Lua[2], Sutthiphong Srigrarom[2], and Boo Cheong Khoo[2]

[1] University of Shanghai for Science and Technology, Shanghai, China
obmmd_zxd@163.com
[2] National University of Singapore, Singapore, Singapore

Abstract. Continuous Sign Language Recognition (CSLR) is a challenging task in the field of action recognition. It requires splitting a video into an indefinite number of glosses, which belong to different classes. Nowadays, researchers usually use deep learning methods with end-to-end training. One popular CSLR model paradigm is a three-step network, *i.e.*, using a visual module to extract 2D frame features and short-term sequential features, then using a sequential module to analyze contextual associations, and finally Connectionist Temporal Classification (CTC) loss is used to constrain the output. Gloss alignment ability is found to be an important factor affecting CSLR model performance. However, the three-step CSLR paradigm mainly depends on the sequential module to align gloss, the visual module only focuses on local information and contributes little to module alignment ability, leading to training inconsistent between these two modules. This paper proposes an Attention Auxiliary Supervision (AAS) method to optimize the parameter of visual module and help it pay more attention to global information, thereby improving the alignment ability of the whole model. As an external part of the main model, the proposed AAS method has flexible applicability and is expected to be used in other CSLR models without increasing the cost of inference. The model performs well on two largescale CSLR datasets, *i.e.*, PHOENIX14 (21.1% Test) and PHOENIX14-T (20.9% Test), which demonstrates its competitiveness among state-of-the-art models.

Keywords: Continuous Sign Language Recognition · Attention · Auxiliary supervision

1 Introduction

Sign language is a communication tool used by deaf-mute people in daily life. It is different from the spoken language we usually use. Spoken language is transmitted by sound, while sign language is transmitted by light and shadow. Due to different modes of transmission, this increases the cost of people learning sign language, resulting in fewer hearing people mastering sign language. Therefore, the study of Sign Language Recognition (SLR) technology is necessary to help normal people to communicate with deaf-mute people. SLR is a transdisciplinary research topic, which involves computer

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 16–28, 2024.
https://doi.org/10.1007/978-981-99-7022-3_2

vision (CV) and natural language processing (NLP). SLR can be divided into Isolated word Sign Language Recognition (ISLR) and Continuous Sign Language Recognition (CSLR). ISLR only needs to classify single action, while CSLR needs to deal with multiple consecutive actions, which are bound to align in addition to classify. Therefore, CSLR research is not only challenging, but also meaningful.

Recently, deep learning has achieved great success in video analysis, and some studies have begun to use deep convolutional neural networks (CNNs) to solve CSLR task [1, 2]. Bi-directional Long Short-Term Memory (BiLSTM) [3] and transformer [6] are used for sequence processing. Some non-end-to-end methods [12, 14], which require an additional fine-tuning process, are proposed to greatly improve the performance of CSLR model. In order to adapt to CSLR task and promote WER performance, Graves *et al.* [4] presents Connectionist Temporal Classification (CTC), which is an efficient method to align predictions with tags for sequences of different lengths.

Obviously, the gloss alignment ability is very important for CSLR tasks, where the input and output are both sequential data. The visualization results of feature similarity matrix show that visual module could learn little about alignment information. In addition, some research [11] has found the convergence rate of the visual module is slower than that of the sequential module. As a result, even when the sequential module is overfitting, the visual module still may not get effective feedback or adequate training. In this regard, a method called Attention Auxiliary Supervision (AAS) is proposed in this paper to enhance the supervision of visual module, thereby enabling it to learn alignment ability in advance, and enhancing the connection between visual module and sequential module. AAS uses a weakly supervised approach to avoid the high computing cost of strong supervision, such as facial expression, hand segmentation, body skeleton, *etc*. Furthermore, AAS is an end-to-end network, which is more elegant than iterative training. The research results are summarized as follows:

- By combining the attention mechanism with auxiliary supervision, the AAS method is proposed to optimize the parameter distribution of visual module, thereby enhancing its constraint capability for gloss alignment.
- AAS module has simple structure and flexible applicability, which is only used during training, so that model performance can be improved without changing the inference structure. The proposed network is validated on PHOENIX14 and PHOENIX14-T, and achieves the best WER (Test) based on PHOENIX14-T.

2 Related Works

2.1 Mechanism of Attention

The attention mechanism is widely used in current deep learning models, and plays an important role. In earlier times, Bengio *et al.* [5] proposed the classical attention mechanism (Bahdanau attention), which was used in the field of NLP. Vaswani *et al.* [6] proposes the structure of transformer that involves self-attention and multi-head attention. In addition, there are many other basic attention variants, such as hard attention [15], local attention [16]. The attention mechanism has also developed much in the CV field, such as ViT [17], Swin-T [18], DETR [19], in which images are usually divided into sequences and then fed into attention operation.

Different from CNN, which extracts local information, the attention mechanism extracts global information, making it suitable for long-distance modeling analysis of sequences. Therefore, its computational efficiency is much higher than that of CNN and RNN, and it can even replace them under certain circumstances, which also makes a great contribution to the research on multi-modal task. The experiment results in this paper show that the attention mechanism is helpful for gloss alignment. Therefore, cross-attention method is used to extract and integrate the information associated with the visual module and sequential module, so that the visual module can learn the proper gloss alignment ability by backpropagation in advance.

2.2 Auxiliary Supervision

Due to the deepening of the neural network layers, the feature transparency of the middle layer of the network is low, and the phenomenon of gradient vanishing or exploding is accompanied, which hinders the network training and fitting. Lee *et al.* [7] introduces the concept of deep supervision as a training trick. This method adds an auxiliary classifier to some middle layers of the deep neural network as the intermediate output. In 2014, GoogLeNet [9] used multiple fully connected layers as auxiliary classifier to conduct auxiliary supervision of the network. In 2015, Wang *et al.* [8] attempts to use this supervision technique in a deeper structured network. More and more works have started to add branch network and train the main network with auxiliary supervision.

Each module in CLSR network has its own function, and direct supervision and training of a specific module can effectively inform its learning task. However, there are progressive dependency relationships between different modules, and complex and redundant network structures can hinder the transmission of information. So, supervising the modules separately can not make the network learn the distributed features efficiently. To solve these problems, this paper designs an auxiliary supervision method that has a simple and straightforward structure, which can enable modules to learn from each other.

3 Our Approach

The proposed network structure is shown in Fig. 1. It is mainly divided into four modules, which are visual module, sequential module, distillation module and AAS module. The network construction including these modules will be described in detail next, through four sections.

3.1 Main Stream Design

As shown in Fig. 1, the input data is $X = \{x_i\}_{i=1}^{T} \in \mathbb{R}^{T \times c \times h \times w}$, which represents T frames, and the value of T is indeterminate. Through the processing of main stream neural network, comes out $C_S \in \mathbb{R}^{\tilde{T} \times (|\mathbb{Q}|+1)}$, where $\mathbb{Q}$ is the gloss dictionary space generated by the data set, and $|\mathbb{Q}|$ is gloss dictionary space length. To align the output with label $L = \{l_i\}_{i=1}^{n} \in \mathbb{Q}$, beam search strategy [27] is used to get the most likely N gloss fragments $Y = \{y_i\}_{i=1}^{N} \in \mathbb{Q}$.

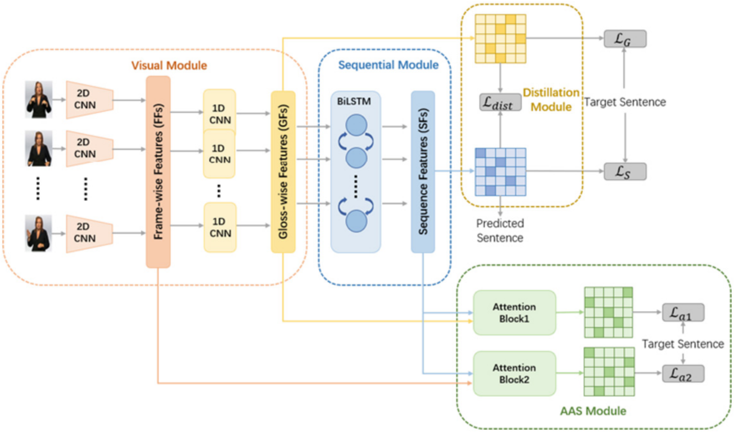


Fig. 1. Overview of the proposed network. Network structure is mainly divided into four modules, which are visual module, sequential module, distillation module, and AAS module. Only visual module and sequential module participate in the inferencing process.

Visual Module. A pre-trained 2D CNN is used to extract image features for each video frame in $X = \{x_i\}_{i=1}^{T} \in \mathbb{R}^{T \times c \times h \times w}$, and then obtain the Frame-wise Features (FFs) $F \in \mathbb{R}^{T \times C}$. That is, an image with a size of $c \times h \times w$ is converted to a C dimensional vector:

$$F = \{f_i\}_{i=1}^{T} = \{\mathcal{C}_{2d}(x_i)\}_{i=1}^{T} \in \mathbb{R}^{T \times C} \tag{1}$$

The features of each frame in FFs are independent of each other. Since the recognition of action needs to consider the temporal relationship, the relationships between the adjacent frame features are needed to be established. This paper uses conventional 1D CNN to extract the temporal features, then the Gloss-wise Features (GFs) are obtained:

$$G = \{g_i\}_{i=1}^{\tilde{T}} = \mathcal{C}_{1d}\left(\{f_i\}_{i=1}^{T}\right) \in \mathbb{R}^{\tilde{T} \times C} \tag{2}$$

Dimension T is downsampled to get $\tilde{T}$, $T > \tilde{T}$. The GFs at this point are visual features containing local temporal information, with classification and a little alignment effect.

Sequential Module. This module uses a two-layer BiLSTM to model the long time relationships in GFs, so that the module could analyze the long time semantic information and obtain the Sequential Features (SFs) $S \in \mathbb{R}^{\tilde{T} \times C}$. Through the classifier, the output $C_S \in \mathbb{R}^{\tilde{T} \times (|\mathbb{Q}|+1)}$ is obtained:

$$S = \{s_i\}_{i=1}^{\tilde{T}} = \mathcal{R}_{Bi}\left(\{g_i\}_{i=1}^{\tilde{T}}\right) \in \mathbb{R}^{\tilde{T} \times C} \tag{3}$$

$$C_S = \{c_i\}_{i=1}^{\tilde{T}} = WS^{\tilde{T}\times C} + b \in \mathbb{R}^{\tilde{T}\times(|\mathbb{Q}|+1)} \tag{4}$$

The '1' in $|\mathbb{Q}| + 1$ represents the space occupied by the blank class generated by the CTC algorithm.

Alignment and Output Sentence. C_S, after *softmax* processing, contains repetitive glosses that are much longer than their corresponding label. So, CTC loss is used to constrain C_S to align it with label, and loss $\mathcal{L}_S$ is obtained. The final result $Y = \{y_i\}_{i=1}^{N} \in \mathbb{Q}$ is obtained through the beam search strategy, which calculates the probabilities of all possible hypotheses at each time slice and selects the highest few as a group. It is an iterative process and stops when the last time slice is reached.

3.2 CTC Loss for Alignment

Since the input and output have different lengths, the corresponding labels have different lengths, which causes CSLR model to output multiple duplicate classes. These duplicate values may represent single class or multiple classes, where some necessary blank items are needed to be generated by classifier. Combining the blank class with the original gloss vocabulary $\mathbb{Q}$ generates a new gloss vocabulary $\widehat{\mathbb{Q}} = \mathbb{Q} \bigcup \{blank\}$. In order to explain the effect of blank items, this paper uses a speech recognition example that outputs sequence $\mathcal{B}(hhelllo) = hello$. If one merges the duplicate values as per the original strategy, it will be processed as *helo*. If proper blank items are generated by classifier, the output sequence might become $\mathcal{B}(hh \cdot ell \cdot lo)$. Merge the repeated items and then the correct result *hello* is obtained.

For the same label, there might exist multiple correct output sequences (paths) with blank items at different positions, that is multiple paths $\pi \in \widehat{\mathbb{Q}}^{T}$ can represent the same label, such as $\mathcal{B}(hh \cdot ell \cdot lo\cdot) = \mathcal{B}(\cdot h \cdot ell \cdot l \cdot oo\cdot) = hello$. CTC calculates the probabilities of all possible paths:

$$p(\pi|X) = \prod\nolimits_{t=1}^{\widehat{T}} p(\pi_t|X), \pi \in \mathcal{B}^{-1}(l) \tag{5}$$

$\mathcal{B}^{-1}(l)$ represents all the possible paths. Then minimize their negative logarithmic likelihood sum to get the CTC loss:

$$p(l|X) = \sum\nolimits_{\pi\in\mathcal{B}^{-1}(l)} p(\pi|X) \tag{6}$$

$$\mathcal{L}_{CTC} = -logp(l|X) \tag{7}$$

3.3 Knowledge Distillation for CSLR

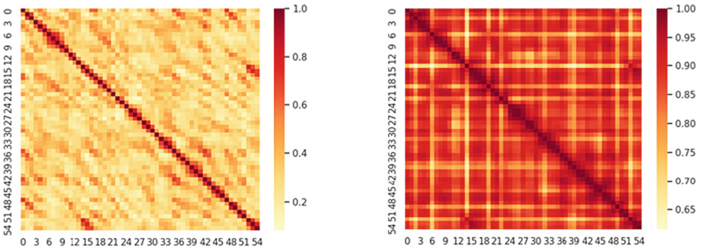
Distillation network is a good approach as it can effectively transfer teacher's knowledge to student, and build a lightweight and efficient student network. Zhang *et al.* [10] proposed a self-distillation network to simplify an otherwise complex structure by enabling the network to learn from itself without the help of external information. Min *et al.* [11]

modified the self-distillation network for CSLR tasks, which is also utilized in this paper to improve effectiveness of model.

In order to enhance the classification capability of visual module, an auxiliary classifier is added directly after GFs to obtain $C_G \in \mathbb{R}^{\tilde{T}\times(|\mathbb{Q}|+1)}$, and generate corresponding CTC loss $\mathcal{L}_G$. However, the lack of context capability makes it difficult for C_G to align with the label. This paper treats sequential module as teacher network and visual module as student network to make GFs learning feature distribution more efficiently. Take the value of C_S as the learning target (soft target) of C_G, calculate the divergence between them as distillation loss:

$$\mathcal{L}_{dist} = KL\left(softmax\left(\frac{C_S}{\tau}\right), softmax\left(\frac{C_G}{\tau}\right)\right) \tag{8}$$

where τ represents the distillation temperature and plays a smoothing role in the output of the network. A relatively large τ value is used to soften the peak probability and smooth the distribution of labels.



(a)

(b)

Fig. 2. Self-similarity matrix heat map. A sample from the PHOENIX14 dataset (01April 2010 Thursday heute default-1) is used as input to the model. (a) is generated from GFs in baseline. (b) is generated from the SFs in baseline.

3.4 Attention Auxiliary Supervision

In Fig. 2(a), the similarity between adjacent frames in GFs is large, but there is no clear boundary between glosses, *i.e.*, no obvious alignment effect. This indicates that the previous feature extraction process only extracts partial temporal information, but the range of information required for alignment is larger. In Fig. 2(b), the alignment of the SFs is obvious, with the lighter part representing non-blank class and the rest representing blank class. Therefore, it is necessary to design a module that helps visual module focus more on global information to enhance their gloss alignment ability.

As the depth of the network increases, the deeper-layer network tends to 'forget' the information learned by the shallower-layer network, and the shallower-layer network also 'forgets' the task to learn. In other words, there is some information loss in the transmission, and the more layers it is transmitted through, the greater the loss. Cross-attention mechanism [6] has the structure of multi-input and single-output. If the outputs of different modules in CSLR network are fed into cross-attention as inputs, the output of cross-attention which integrates information of different modules is expected to be obtained. As a result, the cross-attention mechanism with supervision has the potential of enhancing module connections.

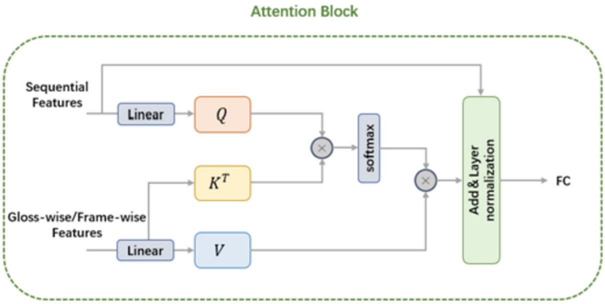


Fig. 3. The internal structure of attention block in AAS module. AAS module includes two attention blocks, whose input consists of two parts. One part is Sequential Features (SFs), the other part is Frame-wise Features (FFs) or Gloss-wise Features (GFs).

This paper designs the Attention Auxiliary Supervision (AAS) module based on cross-attention mechanism as shown in Fig. 3. The AAS module includes two attention blocks with different K and V inputs, *i.e.*, GFs in attention block 1 (att1) and FFs in attention block 2 (att2). Firstly, let SFs pass through the linear layer to get $Q \in \mathbb{R}^{\tilde{T} \times C}$. FFS or GFs are also mapped to K and V via a linear layer (they are the same shape as the FFs or SFs that generate them). Multiply Q with K^T to get the attention score matrix. In order to stabilize the gradient, the attention score matrix is scaled by dividing it with $\sqrt{C}$. Normalize the attention score matrix with *softmax* function, then the attention weight matrix is obtained. Finally, multiply the attention weight matrix with V:

$$Attention_i = softmax\left(\frac{Q \times K^T}{\sqrt{C}}\right) \times V \tag{9}$$

A single attention operation is not sufficient to handle multiple kinds of information, so multiple-head attention is used in AAS module. Concatenate multiple-head attention and combine them through a linear transformation:

$$Attention = cat(Attention_1, \ldots Attention_h)W^O, W^O \in \mathbb{R}^{hc \times c} \tag{10}$$

Regardless of whether FFs or GFs are input, *Attention* always has the same shape as Q, so residuals can be implemented. In order to mitigate gradient vanishing or exploding, layer normalization is carried out:

$$A = LN(Q + Attention) \tag{11}$$

Through the classifier and *softmax*, the outputs of two attention blocks C_{a1} and C_{a2} are obtained. $\mathcal{L}_{a1}$ and $\mathcal{L}_{a2}$ are the CTC losses that are used to constrain the attention outputs. The overall loss of AAS module is:

$$\mathcal{L}_{AAS} = \mathcal{L}_{a1} + \mathcal{L}_{a2} \tag{12}$$

AAS module has not some operation which are contained in original transformer, such as position encoding, weight mask and Feed Forward Neural Network. This is because we believe that these operations will clutter the alignment information and increase the amount of computation. AAS module only plays the role of auxiliary supervision during network training, and this module is ignored when inferencing, without increasing the cost of inference.

The loss function of the whole network consists of four parts, including $\mathcal{L}_S$ generated by SFs, $\mathcal{L}_G$ generated by GFs, $\mathcal{L}_{AAS}$ generated by AAS module, and distillation loss $\mathcal{L}_{dist}$. Because of the different types of loss functions (all belong to CTC loss except $\mathcal{L}_{dist}$), the hyper parameter α is used to balance the loss:

$$L = \mathcal{L}_G + \mathcal{L}_S + \mathcal{L}_{AAS} + \alpha \mathcal{L}_{dist} \tag{13}$$

4 Experiments

4.1 Experimental Setup

Datasets. Two widely used CSLR datasets are adapted to verify the proposed method, i.e., RWTH-PHOENIX-Weather-2014 and RWTH-PHOENIX-Weather-2014-T.

PHOENIX14, the German sign language dataset is built from the 2014 German TV sign language weather forecast. Nine signers generate 6,841 sentences containing 1,295 signs and nearly 80,000 words.

PHOENIX14-T is an extension of PHOENIX14 that can be used to evaluate CSLR and Sign Language Translation (SLT) tasks. It has fewer signs, but more content. The dataset contains 8,247 sentences, and its dictionary consists of 1,085 signs.

Evaluation Metric. Word Error Rate (WER) is widely used in CSLR as an indicator to judge recognition results. The lower WER value is, the better the performance will be. The number of error results can be calculated by summing the number of insertions (#ins), substitutions (#sub), and deletions (#del). WER is the proportion of error results in the whole length of labels (#lab):

$$WER = \frac{\#ins + \#sub + \#del}{\#lab} \tag{14}$$

Implementation Details. The experiment used two Nvidia RTX 3090s. During training, all video frames are randomly cropped into 224 × 224. In order to reduce the boundary effect during the one-dimensional convolution, the first frame and the last frame are padded. The padding length is determined by the degree of one-dimensional convolution. Since the number of frames in the input video is indefinite, all videos in the same batchsize are padded into the same length, which is the maximum number of sample frames in the batchsize. In visual module, pre-trained ResNet18 is used as the backbone to extract image features. ResNet18 has low complexity, and can extract enough image feature information. In the sequential module, a two-layer BiLSTM with a hidden state dimension of 2 × 516 is used. In the distillation module, τ of $\mathcal{L}_{dist}$ is set to 8. Finally, the number of neurons in the classifier is set to $|\mathbb{Q}| + 1$.

4.2 Ablation Studies

In this section, ablation experiment results based on PHOENIX14 are given to verify the effectiveness of the proposed methods. The main stream network introduced in the Sect. 3.1 is designed as the baseline of the experiment, and other modules or methods are added on this basis to compare and demonstrate their effects.

Table 1. Ablation studies of AAS module on the PHOENIX-2014 (att1 connects the GFs and SFs. Att2 connects the FFs and SFs)

Methods	Dev (%)	Test (%)
Base.	23.3	23.8
Base. + att1	20.6	21.9
Base. + att2	21.0	22.3
Base. + att1 + att2	**20.3**	**21.7**

Ablation of Attention Blocks in AAS. As Table 1 shows, both att1 and att2 are effective when used alone. Because FFs and SFs are far away from each other, the information difference between them is large, so the performance of att2 is inferior to that of att1. The best results are achieved when att2 and att1 are used together. Figure 4 is the self-similarity matrix heat map that is generated by the GFs after adding the AAS module when the same input sample is used as Fig. 2. As Fig. 4 shows, obvious alignment effect appears in the self-similarity matrix of GFs when AAS module is used, which demonstrates that the proposed AAS module can enhance the alignment ability of visual module.

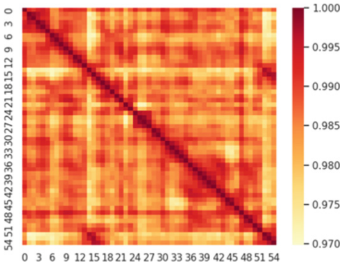


Fig. 4. Self-similarity matrix heat map. It is generated by the GFs after adding the AAS module.

Ablation of Positional Encoding Methods in AAS. In this work, several positional encoding methods are tried like transformer does, with the expectation to further improve the model performance. However, as Table 2 shows, these positional encoding methods do not have positive effect, and even cause model deterioration. The reason might be that CSLR task only needs to align the glosses without adjusting their order. In addition, SFs

Table 2. Effect of positional encoding in AAS, based on PHOENIX-2014. (PPE: Parameter trainable positional encoding. APE: Absolute positional encoding. RPE: Relate positional encoding. NPE: Non-positional encoding)

Methods	Dev (%)	Test (%)
PPE	21.8	23.6
APE	21.7	23.4
RPE	21.4	22.8
NPE	**20.3**	**21.7**

Table 3. Ablation of AAS and distillation module on the PHOENIX14. (Dist.: distillation module)

Base.	AAS	Dist.	Dev (%)	Test (%)
*			23.3	23.8
*		*	21.2	22.3
*	*		20.3	21.7
*	*	*	**19.9**	**21.2**

after BiLSTM contain some positional information, and adding additional positional information may mess up the features and hence increase the training burden.

Table 4. Performance comparison on PHOENIX14 dataset and PHOENIX14-T dataset. $*$ indicate extra cues such as face, hand features or information of other modes. ‘del’ and ‘ins’ stand for deletion error and insertion error, respectively.

Methods	PHOENIX14		PHOENIX14-T	
	Dev (%)	Test (%)	Dev (%)	Test (%)
SubUNet [24]	40.8	40.7	-	-
SFL [23]	26.2	26.8	25.1	26.1
FCN [22]	23.7	23.9	23.3	25.1
CMA [21]	21.3	21.9	-	-
VAC [11]	21.2	22.3	-	-
TLP [13]	**19.7**	**20.8**	**19.4**	**21.2**
C+L+H* [26]	26.0	26.0	22.1	24.1
SLT* [25]	-	-	24.5	24.6
DNF* [12]	23.1	22.9	-	-
STMC* [20]	**21.1**	**20.7**	**19.6**	**21.0**
AAS (ours)	**19.9**	**21.1**	**19.5**	**20.9**

Ablation of AAS and Distillation Module. The proposed AAS module is inspired by the self-distillation operation [11]; they both belong to auxiliary supervision method, that can facilitate information transmission and optimize parameter distribution. As shown in Table 3, the AAS module outperforms the distillation module when only one of them is used. The reason is, in addition to intermediate information generated by network modules, the AAS module also utilizes the final label information to supervise the training process, while the distillation module only uses self-generated pseudo labels.

Using AAS and distillation modules together can then achieve the best result, which demonstrate these two modules can complement each other.

4.3 Comparison with State-of-the-Arts

As shown in Table 4, the proposed model is compared with other state-of-the-arts models. WERs on PHOENIX14 are 19.9% (Dev) and 21.2% (Test). WERs on PHOENIX14-T are 19.5% (Dev) and 20.9% (Test). The proposed model achieves the best performance on WER (Test) of PHOENIX14-T.

The * in Table 4 represents that the model uses extra cues, including hand, face, or other modal information. Most indicators of the proposed model are better than STMC* [20], which is the best model in *. The proposed model also outperforms most other models using only video information except TLP [13]. The performance of the proposed model is comparable with the current best model (TLP) to our known, and it even outperforms TLP on WER (Test) of PHOENIX14-T. As a pooling method, TLP needs to modify the main network of CLSR. However, as an auxiliary supervision method proposed in this paper, it can maintain the original CSLR main network and is only used during training.

5 Conclusions

The visual module in CSLR network mainly focuses on the 2D spatial features of input frames, and can extract only short-term temporal relationships, which is detrimental to the gloss alignment ability of CSLR network. This paper designs an attention auxiliary supervision network based on the distillation module and the proposed AAS module, which can enhance the information interaction between visual module and sequential module, and enable them to learn gloss alignment ability in advance. The experiment results based on PHOENIX14 and PHOENIX14-T demonstrate the superiority of the proposed network among state-of-the-art models. As a kind of auxiliary supervision method, the AAS module is only active during training, and needs no change of CSLR main network, thereby it is expected to facilitate other sequential related tasks.

Acknowledgements. This work is funded by the "Project research on human–robot interactive sampling robots with safety, autonomy, and intelligent operations" supported by National Natural Science Foundation of China (NSFC), Grant/Award Number: 92048205; This work is also funded by China Scholarship Council (CSC), Grant/Award Number: 202008310014.

References

1. Camgoz, N.C., Hadfield, S., Koller, O., Bowden, R.: SubUNets: end-to-end hand shape and continuous sign language recognition. In: ICCV, pp. 3075–3084 (2017)
2. Cui, R., Liu, H., Zhang, C.: A deep neural framework for continuous sign language recognition by iterative training. In: TMM, pp. 1880–1891 (2019)

3. Pu, J., Zhou, W., Li, H.: Iterative alignment network for continuous sign language recognition. In: CVPR, pp. 4165–4174 (2019)
4. Graves, A., Fernández, S., Gomez, F., Schmidhuber. J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv pre-print arXiv:1409.0473 (2016)
6. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
7. Lee, C.-Y., Xie, S., Gallagher, P.W., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Proceedings of the Artificial Intelligence and Statistics (2014)
8. Wang, L., Lee, C.-Y., Tu, Z., Lazebnik, S.: Training deeper convolutional networks with deep supervision. arXiv preprint (2015)
9. Szegedy, C., et al.: Going deeper with convolutions. arXiv preprint (2015)
10. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be your own teacher: improve the performance of convolutional neural networks via self distillation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3713–3722 (2019)
11. Min, Y., Hao, A., Chai, X., Chen, X.: Visual alignment constraint for continuous sign language recognition. In: ICCV (2021)
12. Cui, R., Liu, H., Zhang, C.: A deep neural framework for continuous sign language recognition by iterative training. IEEE Trans. Multimedia **21**, 1880–1891 (2019)
13. Hu, L., Gao, L., Liu, Z., Feng, W.: Temporal lift pooling for continuous sign language recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. ECCV 2022. LNCS, vol. 13695, pp. 511–527. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19833-5_30
14. Pu, J., Zhou, W., Li, H.: Iterative alignment network for continuous sign language recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4165–4174 (2019)
15. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: NeurIPS (2014)
16. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
17. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale learning. In: CVPR (2020)
18. Liu, Z., et al.: Swin Transformer: hierarchical vision transformer using shifted windows. In: CVPR (2021)
19. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
20. Zhou, H., Zhou, W., Zhou, Y., Li, H.: Spatial-temporal multi-cue network for continuous sign language recognition. In: AAAI (2020)
21. Pu, J., Zhou, W., Hu, H., Li, H.: Boosting continuous sign language recognition via cross modality augmentation. In: ACM MM (2020)
22. Cheng, K.L., Yang, Z., Chen, Q., Tai, Y.-W.: Fully convolutional networks for continuous sign language recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12369, pp. 697–714. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_41
23. Niu, Z., Mak, B.: Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12361, pp. 172–186. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58517-4_11

24. Camgoz, N.C., Hadfield, S., Koller, O., Bowden, R.: SubUNets: end-to-end hand shape and continuous sign language recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3075–3084 (2017)
25. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7784–7793 (2018)
26. Koller, O., Camgoz, N.C., Ney, H., Bowden, R.: Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallel-ism in sign language videos. In: PAMI (2019)
27. Hannun, A.: Sequence modeling with CTC. Distill **2**(11), e8 (2017)

AutoShape: Automatic Design of Click-Through Rate Prediction Models Using Shapley Value

Yunfei Fang[1], Caihong Mu[1], and Yi Liu[2](✉)

[1] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Collaborative Innovation Center of Quantum Information of Shaanxi Province, School of Artificial Intelligence, Xidian University, Xi'an 710071, China

[2] School of Electronic Engineering, Xidian University, Xi'an 710071, China

yiliuxd@foxmail.com

Abstract. As a main part of automated machine learning, Neural Architecture Search (NAS) has been employed in exploring effective Click-Through Rate (CTR) prediction models of recommender systems in recent studies. However, these studies update the architecture parameters by different strategies, such as gradient-based methods or evolutionary algorithms, and may suffer from either interference problems or high time-consuming in the search stage. Besides, the blocks in the search space were usually homogeneous, which means they have the same candidate operations to choose. Such design will result in redundancy in the search space, because many structures are inherently invalid and just increase the complexity of searching. To address the above issues, we implement the three-level automatic design of CTR prediction models using NAS via Shapley value, named as AutoShape. For the search space, we divide it into three parts according to the characteristics of the CTR model. Each part comprises distinctive candidate operations and forms a cell as an individual processing level, which improves the stability of the searched models' effect and reduces the computational complexity. For the search strategy, we leverage Shapley value, a metric derived from cooperative game theory, to estimate the contribution of the operations in each block and the connections between the blocks, which can find effective models and reduce the time cost. Furthermore, experiments on different benchmark datasets show that the structure obtained from the search performs better than several hand-crafted architectures and is more stable than another NAS-based algorithm.

Keywords: Neural architecture search · Click-through rate prediction · Shapley value

1 Introduction

In recent years, the development of deep neural networks has significantly influenced the Click-Through Rate (CTR) prediction in Recommender Systems (RSs). The success of CTR prediction architectures depends on their ability to capture and exploit effective low-order and high-order feature interactions [1]. By incorporating and learning from the interactions of different features, models can gain a more general and comprehensive

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 29–40, 2024.
https://doi.org/10.1007/978-981-99-7022-3_3

understanding of the explicit relations between the features of users and items, leading to enhanced performance. However, the choice of operation is crucial as simple operations may lead to information loss, while complex operations can introduce noise or increase the complexity. Consequently, the integration of explicit and implicit interactions remains an underexplored area that extensive manual efforts are required to find the most suitable structure for each scenario, making it difficult to design an efficient CTR model tailored to various tasks.

To overcome these challenges, recently, there has been a growing interest in automating the components of Deep Recommender Systems (DRS) due to the advancements in Automated Machine Learning (AutoML) [2], especially applying Neural Architecture Search (NAS) approaches to design CTR models automatically. NAS has been successfully applied in various computer vision tasks, and the automatically searched architectures have achieved highly competitive performance in image classification and object detection. However, directly applying NAS methods from the computer vision field to the CTR prediction in RSs poses two main challenges, i.e., the redundancy of the search space and the instability of the search strategy. Specifically, it is studied and pointed out that the search space in NAS often contains redundancy increasing the complexity of the search process but not significantly contributing to performance improvements [3]. This is a serious problem and worth studying, especially for the CTR models very sensitive to noise information, because it will lead to a large number of ineffective interaction blocks in the search space, resulting in the very low efficiency of a large number of structures and the highly unstable structure searched in each iteration. Besides, the search strategy also plays a key role in the whole search process. Most of the previous studies using NAS to design CTR architectures or feature interactions mainly adopt gradient method [1, 2, 4, 5] to update architecture parameters. But there exists interference problem that makes the search unstable, because the shared operators in the weight-sharing NAS methods receive different gradient directions from child models with distinct architecture during optimization, which leads to a poor ranking correlation between the estimated accuracy and the ground truth accuracy.

To address the above issues, we implement the three-level automatic design of CTR prediction models using NAS via Shapley value, which is named as AutoShape. Considering the redundancy problem of search space, we construct the search space consisting of three parts with different operations according to the general characteristics of existing CTR prediction models. To be more specific, the overall search space consists of three different cells, and the blocks in each cell contain different operations to extract useful information. These blocks are used to extract important features from the raw input, perform explicit interactions at the vector level, and perform implicit interactions at the bit level. Finally, we use multi-head attention mechanism to capture important output information for prediction. This design follows the prior knowledge about the basic process of feature processing in CTR prediction models, and narrows down the search space by reducing the number of ineffective structures. It is worth pointing out that current NAS-based work primarily emphasizes the efficiency of searching for the optimal structure, without extensively investigating the stability of efficiency for each search iteration. Our design significantly improves the search stability, which was neglected by previous NAS-based approaches.

To address the interference issue in gradient-based search, we adopt a novel search strategy using the Shapley value, which is a metric derived from cooperative game theory. The Shapley value provides a straightforward way to assess the importance of operations and edges in the supernet by analyzing the variations of validation accuracy. We consider the relationships between blocks that composed of different operations by the average marginal contribution. This approach is effective in obtaining the importance of the candidate operations as architecture parameters and does not require gradient computations through backpropagation, thereby reducing interference problems. Shapley-NAS [6] first introduced Shapley-value in the field of computer vision to compute the contributions of each operation and edge in the search space for NAS. While our work is the application of the Shapley value in NAS for the automatic design of CTR prediction models in RSs.

By leveraging this innovative approach, we aim to overcome the challenges posed by interference and improve the effectiveness of the search process for the automatic design. The main contributions are as follows:

– We propose a staged and refined search space for CTR models, which aligns with the characteristics of CTR prediction and ensures the majority of structures are effective, thereby improving search stability and reducing redundancy.
– We are the first to adapt the Shapley value from the NAS for computer vision to the NAS for RSs. We propose a single-block sampling strategy to evaluate the architecture parameters via Shapley value based on the structure of CTR prediction models.
– We employ the multi-head attention mechanism to corresponding weights of each block, facilitating the selection of relevant output information and reducing interference from noisy signals.
– Experimental results demonstrate that our algorithm achieves higher efficiency compared to manually designed models and exhibits greater stability.

2 Related Work

CTR prediction plays a significant role in predicting user engagement and enhancing recommendation quality. Factorization Machine (FM) [7], as a classical feature interaction method, captures feature interactions through vector inner products. In recent studies, there has been a growing emphasis on exploring deep learning methods to enable effective interactions among diverse feature types, such as Dot Product [8], Cross Network (CN) [9], Compressed Interaction Network (CIN) [10], and Bilinear-based [11, 12]. These works usually put efforts into designing explicit feature interactions, which are combined with implicit interactions.

AutoML has become a widely adopted approach for exploring and identifying suitable model structures. There has been increasing attention on the application of AutoML in RSs in recent years. AutoCTR [13] pioneered the automatic search for effective architectures in CTR prediction. AutoPI [4] employed a gradient-based search strategy to enhance computational efficiency. To further reduce the computational complexity of search, NAS-CTR [5] proposed a differentiable NAS approach based on proximal iteration to jointly optimize the network weights and architecture parameters.

3 Method

3.1 Search Space of AutoShape

For the design of the search space, we partition the space into three parts considering the feature characteristics of input, interaction, and output in CTR models. Each stage corresponds to a cell, namely Input Cell, Inter Cell, and MLP Cell. The blocks within each cell exclusively contain operations specific to that stage.

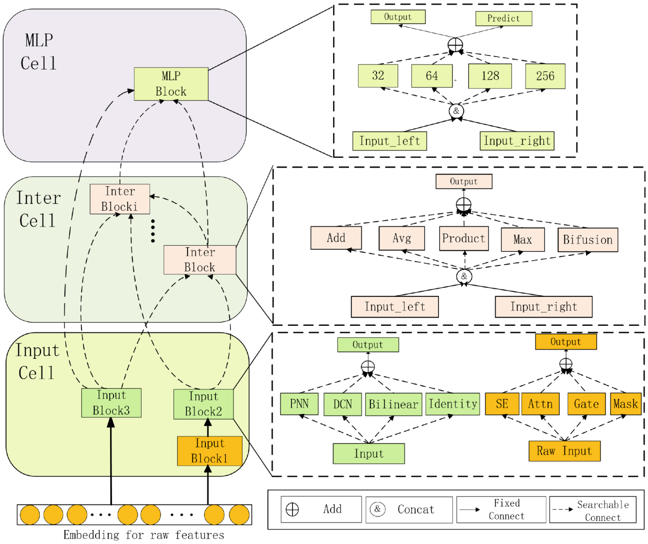


Fig. 1. Schematic diagram of three-level search space.

Figure 1 illustrates the three-level search space of AutoShape. The initial input is the embedding representation of the features, where the high-dimensional sparse features are embedded using an embedding table, resulting in each feature being a low-dimensional embedding vector. We define $C = [C_1, C_2, C_3]$ that contains three cells in the search space. $B_i = [B_i^1 \ldots, B_i^{m^i}]$ describes the blocks in the i-th cell, where $1 \leq i \leq 3$, and m^i is the total number of blocks in the i-th cell. $O_i = [O(B_i^1) \ldots, O(B_i^{m^i})]$ enumerates the set of operations, and $O(B_i^j)$ includes all operations of block B_i^j, $1 \leq j \leq m^i$. . The Input Cell (i.e., C_1) primarily focuses on extracting relevant information from the raw embeddings and enabling feature interactions among vectors. C_1 consists of three blocks, each containing different operations. B_1^1 is used to learn and select the importance of each feature interaction. It includes operations such as Multi-Head Attention (Attn) [14], SENet (SE) [12], Gating and Mask. These operations enable the selection of salient latent information from the original embedding vectors for further feature processing. The operations in B_1^2 and B_1^3 perform feature interactions among the information derived from B_1^1 and the original feature embeddings to extract explicit information, including CrossNet [9], PNN [8], Bilinear [12] and Identity operation. Suppose X_i^j and Y_i^j represent the input and output information of block B_i^j respectively, and X_1^0 is the embedding for raw features. The information processing flow of blocks in C_1 is depicted by Eq. (1):

$$Y_1^j = \sum_{o \in O(B_1^j)} \left(X_1^{j-1} \right) \tag{1}$$

The Inter Cell (i.e., C_2) allows each block to receive the outputs from the previous two blocks. The blocks in C_2 are designed similarly to EDCN [15], which enables the concatenation of two previous tensors and C_2 performs interactive operations among the bits, serving as a bridge. The operations in include summing (Add), averaging (Avg), taking the maximum value (Max), Hadamard product (Product), and Bi-linear operation (Bifusion) between the two input feature vectors. The information processing flow of blocks in C_2 is shown as Eq. (2):

$$Y_2^j = \sum_{o \in O(B_2^i)} o\left[\mathrm{Concat}\left(Y_{i_1}^{j_1}, Y_{i_2}^{j_2} \right) \right] \tag{2}$$

Both $Y_{i_1}^{j_1}$ and $Y_{i_2}^{j_2}$ are the output information of blocks $B_{\mathrm{i}_1}^{j_1}$ and $B_{\mathrm{i}_2}^{j_2}$, which are blocks in front of B_2^j. The two output tensors are concatenated, and then the operations existing within the block are applied to the input features, and the aggregated information is used as the output of this block.

C_3 is referred as the MLP Cell because the blocks in it only contain MLP operation. Common CTR prediction models typically place the MLP layer after or in parallel with the feature interaction layer, enabling the extraction of implicit interaction information from features. The processing flow of blocks in the MLP Cell can be shown as Eq. (3):

$$Y_3^j = \sum_{d \in Dim} \mathrm{MLP}^{(d)}\left[\mathrm{Concat}\left(Y_{i_1}^{j_1}, Y_{i_2}^{j_2} \right) \right] \tag{3}$$

Dim is a set of numbers of hidden layer units. $\boldsymbol{MLP}^{(d)}$ represents an MLP network with d neurons in the hidden layer, where $d \in Dim$. This allows for searching different MLP architectures by varying the number of hidden neurons and finding an appropriate scale for prediction.

Compared to the search space of NAS-CTR, AutoShape divides the original space into three parts according to the characteristics of different stages of feature information processing, reducing the structure redundancy and improving the search stability.

3.2 Search Strategy

In cooperative game theory, a group of players collaboratively form a coalition. Shapley value is used to allocate the payoffs in cooperative games, considering the contributions of participants in different coalitions. Specifically, the Shapley value indicates the relative contribution of one player by calculating the average marginal contribution of this player over all possible coalitions.

During the process of using NAS methods to search for CTR models, AUC (Area Under the ROC Curve) is used as the metric to measure the performance of a CTR model searched in the supernet. We consider that all candidate operations O in the supernet and the edges E between the blocks as players. All players collectively form a coalition $P = O \cup E = \{p^{(1)}, p^{(2)}, \ldots\ldots, p^{(n)}\}$, where $|P| = n$, which is the total number of players and $p^{(i)}$ is the i-th player, i.e., one operation of one block or an edge. Here we use the AUC on the validation dataset as the validation accuracy V to evaluate the contribution of one player in P. $\phi_p^{(i)}$ is the Shapley value of $p^{(i)}$. To obtain $\phi_p^{(i)}$, first, we figure up all possible subnets of the coalition without $p^{(i)}$, denoted as $P\backslash\{p^{(i)}\}$ and S is one of them, i.e., $S \in P\backslash\{p^{(i)}\}$. . Then for the network Net(S) corresponding to S, we calculate the difference in the validation accuracy of Net(S) with $p^{(i)}$ before and after removing it, which is taken as the marginal contribution of $p^{(i)}$ over subset S. Finally, we sum up all the marginal contribution of $p^{(i)}$ over different S and normalize them as the Shapley value of $p^{(i)}$. Therefore, the Shapley value for the player $p^{(i)}$ is calculated by Eq. (4):

$$\phi_p^{(i)}(V) = \frac{1}{|P|} \sum_{S \subseteq P\backslash\{p^{(i)}\}} [V(S \cup \{p^{(i)}\}) - V(S)]/C_{|P|-1}^{|S|} \tag{4}$$

As the number of candidate operations $|O|$ and the edges $|E|$ in the supernet is large, obtaining the Shapley value will become extremely difficult, which requires about $2^{(|O|+|E|)}$ calculations. Therefore, to reduce the evaluation time, a Monte Carlo sampling method is employed to estimate the Shapley value. Monte Carlo sampling method estimates the Shapley value by generating some random permutations of player orders, rather than computing marginal contributions through all the permutations. However, the order of calculating player contributions in each permutation will have a significant impact on the efficiency of the model, as the removal of specific edges will affect the integrity of the supernet. Through extensive analysis and experimentation, we propose a Single-Block sampling strategy, which can obtain effective permutation sequences at the permutation generation stage of Monte Carlo sampling method.

During the sampling, to avoid generating invalid networks, as shown in Fig. 1, we only consider the subset S corresponding to the network that contains all suggested blocks, and each block only has one operation with only specified number of edges. After multiple samplings for all the players in P, we will obtain the Shapley values for all the players, which are the operations and edges within or between different blocks. That means there will exist multiple prominent operations and edges with the same highest Shapley values related to one block. These prominent operations and edges related to each block are picked out to form a downsized set, over which the Shapley values of the elements in it are calculated in the same way as before. The operation and the edge related to each block with the highest Shapley values in that block are selected to form the final network.

This strategy samples among different blocks evenly in the initial stage, and concentrates the sampling on elements with higher Shapley values later, which enables faster and more accurate identification of significant operations and edges.

3.3 Training and Prediction of Supernet

We adopt the One-Shot NAS approach, which is one of the most promising methods in the current field of NAS. The core of this method is training a weight-sharing supernet, where the entire search space is represented by a computation graph that includes all possible operations. Any potential architecture within the search space can be obtained by sampling paths in the supernet graph.

Two types of parameters need to be updated and optimized in the supernet: network parameters w and architecture parameters α. We conduct the pre-training of the supernet on the training set, primarily focusing on optimizing the network parameters w. Then, for the pre-trained supernet, we use the sampling method mentioned in Sect. 3.2 to compute the Shapley value for each operation and edge. The Shapley values of all candidate operations and the edges obtained through multiple samples are used as the architecture parameters α. Among all candidate operations and edges of each block, we select both of them with the highest Shapley value as the final operation and edge.

For the final prediction of the model, we combine the output information from different cells to form N_{out} output modules of dimension d for the final prediction. As the outputs of these blocks are generated by different operations interacting with different blocks, some outputs are effective while others contain noise. Therefore, selecting and extracting useful output information while suppressing noise can effectively improve prediction accuracy. We utilize attention mechanism to extract useful information from these d-dimensional vectors for the final prediction, as shown in Eq. (5):

$$u_i = \text{Attention}(Y_i) = \sum_{j=1}^{N_{\text{out}}} a_i^{(j)} Y_j \tag{5}$$

Here a_i^j denotes the attention of output information between Y_i and Y_j. The output vector Y_i of the original i-th block is weighted and fused through the attention mechanism to obtain u_i. The fused vectors are then concatenated and flattened into $D_{\text{out}} \in R^{1\times(N_{\text{out}}\times d)}$. Finally, a linear transformation, the sigmoid operation is applied as one part of the final prediction value. The other part of the prediction value is composed of the output from the MLP Cell, as shown in Eq. (6):

$$y = \text{Sigmoid}(\text{Linear}(D_{\text{out}})) + \text{MLP}_{\text{out}} \tag{6}$$

The Logloss is adopted as the loss function to train the model, defined as Eq. (7):

$$\text{Logloss} = -\frac{1}{N}\sum_{i=1}^{N}\big(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\big) \tag{7}$$

where y_i and $\hat{y}_i$ are the ground truth of user clicks and predicted CTR, respectively, and N is the number of training samples.

4 Experiments and Results

4.1 Experiment Setting

In this paper, we use two commonly used datasets for CTR prediction, including Avazu and Frappe. The information about these datasets is shown in Table 1:

Table 1. Statistics of datasets.

Datasets	Samples	Fields	Features
Avazu	40,428,967	23	2,018,003
Frappe	288,609	10	5382

Each dataset is split into three parts: D_{train} (80%), D_{val} (10%), and D_{test} (10%). D_{train} is used to train the network parameters w, D_{val} is used to compute the Shapley Value and update the architecture parameters α, and the structure obtained through supernet search is tested on D_{test}. The overall search space consists of three cells with a total of seven blocks. The first stage is the Input Cell with three blocks, followed by the Inter Cell with three blocks as the second stage, and the final stage is the MLP Cell with one block. The output dimensions between adjacent blocks are set to 400, and the embedding size is 16. The Adam optimizer is used to optimize the network parameters.

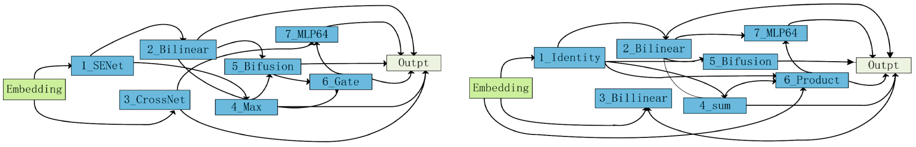
4.2 Performance Comparison

To validate the effectiveness of the CTR model obtained through AutoShape search, we compare it with the human-crafted models including: Logistic Regression (LR), FM [7], Wide & Deep (WD) [16], Deep & Cross Network (DCN) [9], Product-based Neural Networks (IPNN) [8], Deep Factorization Machine (DeepFM) [17]. For the NAS-based method, we compare the best architectures of NAS-CTR provided in the original paper [5], which only included the Avazu dataset. AUC is to measure the performance of CTR models. LogLoss is adopted as the loss function. Each model is subjected to three tests, and the average value is calculated as the final results.

Table 2. Comparison with baseline models.

Dataset	Metric	Human-Crafted Models						NAS-method	
		LR	FM	WD	DCN	IPNN	DeepFM	NAS-CTR	Auto-Shape
Avazu	AUC ↑	0.7563	0.7766	0.7782	0.7799	0.7879	0.7823	0.7867	**0.7885**
	Loss ↓	0.3928	0.3914	0.3885	0.3826	0.3751	0.3819	0.3765	**0.3732**
Frappe	AUC ↑	0.9367	0.9787	0.9812	0.9803	0.9822	0.9802	0.9820	**0.9822**
	Loss ↓	0.4879	0.1831	0.1853	0.1439	**0.1353**	0.1446	0.1783	0.1762

From Table 2, we can observe that AutoShape performs well on large-scale datasets with a substantial number of features, resulting in improvements compared to some traditionally human-designed models and the NAS-based method. Figure 2 shows the best architecture searched by AutoShape on two datasets. By observing these two structures, we can see that for data with a high proportion of sparse features, selecting the Bilinear operation in the first stage is more suitable for extracting effective features.

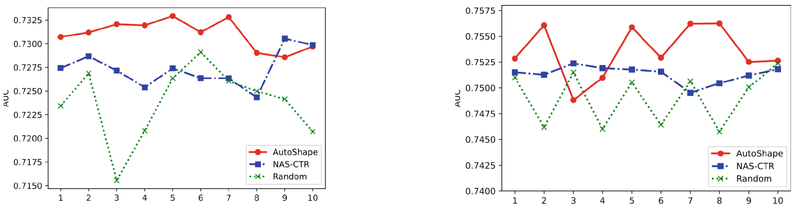


(a) Avazu

(b) Frappe

Fig. 2. The best architecture searched by AutoShape on two datasets.

To validate the stability of the models searched by our methods, we compared it with another NAS-based method NAS-CTR and the random search. To mitigate the interference caused by an excessive presence of a particular feature in the data, we sample 200k and 2 million data from the Avazu dataset. Each method is tested 10 times. The results of 10 times are shown in Fig. 3.



(a) Metrics on 200k

(b) Metrics on 2 million

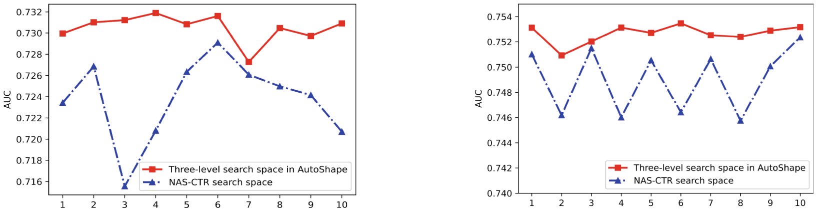
Fig. 3. Experimental results of different NAS-based methods.

From Fig. 3, it can be observed that AutoShape significantly enhances the efficiency of the searched models. This leads to a great improvement in the reliability of the search. This improvement primarily stems from the design of the search space, which reduces the number of ineffective structures. Additionally, the utilization of Shapley value enables the discovery of high-performing structures within the search space.

4.3 Ablation Experiments

Effect of Three-level Search Space. We further investigate the role of the Three-level search space in preserving effective structures. We compare the NAS-CTR search space with the proposed search space in this paper. To reduce the influence of different search strategies on the search results, we adopt the simplest random search on both search spaces and test the structures obtained from 10 searches.

The results in Fig. 4 show that the efficiency of the network searched on the 200k and 2 million datasets, where we adopt the simplest random search on both search spaces (i.e., three-level search space in Autoshape and the NAS-CTR [5] search space) and test the structures obtained from 10 searches. The ten results of the three-level search space are shown in the red line, which has been significantly improved in AUC values



(a) Metrics on 200k

(b) Metrics on 2 million

Fig. 4. Experimental results for stability of different search spaces.

with less fluctuation. However, as for the search space of the original NAS-CTR, due to the existence of many redundant structures in the space, the results of the network structure obtained through search show great instability, as shown in the blue line in the figure. In fact, the design of the first three blocks in the first stage of space searching is crucial because subsequent blocks primarily interact based on the information within these preceding blocks. Therefore, the effectiveness of the output information in the first stage of space searching directly impacts the overall network performance. In our designed search space, the operations within B_1^1 are primarily used to extract relevant portions of features and facilitate feature interaction. In the NAS-CTR search space, the operations corresponding to the initial few blocks in the first stage are randomly selected. Therefore, there is a high probability of selecting ineffective operations that disrupt the valid information within the original embeddings.

Effect of Single-Block Sampling Strategy. During the computation of Shapley value, we employ the Single-Block Sampling Strategy for sampling. In the initial stage, each block selects only one operation, and after multiple samplings, operations with the same contribution are grouped together. The process is repeated until the operation with the maximum contribution is selected. We set the number of samples per iteration to 10 and conduct 10 tests on the 2-million dataset sampled from the Avazu dataset to compare it with the conventional random sampling method.

Table 3. Comparison with different Sampling Strategies.

Sampling method	Average value	Optimum value	Average Time
Single-Block	0.7535	0.7563	448.66
Random	0.7519	0.7553	698.37

As shown in Table 3, the models obtained by Single-Block Sampling Strategy exhibit superior average performance compared to conventional random sampling, with a reduced average time cost. By limiting the single operation sampling within each block in the initial stage, we ensure that each block has at least one operation with its corresponding marginal contribution value. This prevents excessive computation frequency in a single module, which could overshadow the contributions from other modules. In essence, it enhances the global search capability in the early stage and subsequently selects a batch of operations with equal and maximum contributions for calculation, effectively performing local search with promising operations.

Effect of Attention Mechanism. The model concatenates the output information from different types of cells' modules and performs a linear transformation before using it for final prediction. To verify its effectiveness, we randomly select five structures searched on 200k dataset and arrange them in descending order based on their performance. Then, we compare their performance with the performance after incorporating an attention mechanism on output information.

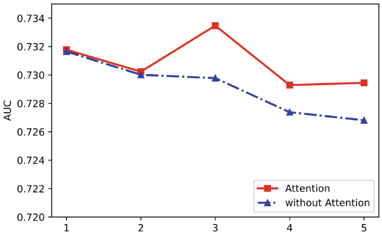


Fig. 5. Effect of attention mechanism.

From Fig. 5, it can be observed that for high-performing models, the attention mechanism has little effect. However, for models with average or poor performance, the attention mechanism can significantly improve the model's performance. This means that for poorly performing architectures, certain blocks may generate noise, resulting in lower performance, while using attention mechanism can reduce the interference of noise and improving the model's performance.

5 Conclusion

We implement the three-level automatic design of click-through rate prediction models by using neural architecture search (NAS) via Shapley value, named as AutoShape. In AutoShape, the search space is narrowed down, which effectively eliminates a significant portion of redundant network structures. The use of the single-block sampling method for computing Shapley value help mitigate interference. Extensive experiments demonstrate that AutoShape is capable of discovering excellent network structures while maintaining good stability. In the future, we will focus on developing lightweight NAS frameworks for recommender systems.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Nos. 62077038, 61672405, 62176196 and 62271374).

References

1. Liu, B., Zhu, C., Li, G.: AutoFIS: Automatic feature interaction selection in factorization models for click-through rate prediction. In: Proceedings of the 26^{th} ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2636–2645 (2020)
2. Zheng, R., Qu, L., Cui, B., et al.: AutoML for deep recommender systems: a survey. ACM Trans. Inform. Syst. (2023)
3. Wan, X., Ru, B., Esperança, P. M., Li, Z.: On redundancy and diversity in cell-based neural architecture search. arXiv preprint arXiv:2203.08887 (2022)
4. Meng, Z., Zhang, J., Li, Y., et al.: A general method for automatic discovery of powerful interactions in click-through rate prediction. In: 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1298–1307 (2021)
5. Zhu, G., Cheng, F., Lian, D., et al.: NAS-CTR: efficient neural architecture search for click-through rate prediction. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 332–342 (2022)
6. Xiao, H., Wang, Z., Zhu, Z., et al.: Shapley-NAS: discovering operation contribution for neural architecture search. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11892–11901 (2022)
7. Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining, pp. 995–1000 (2010)
8. Qu, Y., Cai, H., Ren, K.: Product-based neural networks for user response prediction. In: 16th international conference on data mining (ICDM), pp. 1149–1154 (2016)
9. Wang, R., Fu, B., Fu, G., et al.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD 2017, pp. 1–7 (2017)
10. Lian, J., Zhou, X., Zhang, F., et al.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1754–1763 (2018)
11. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 355–364 (2017)
12. Huang, T., Zhang, Z., Zhang, J.: FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In:13th ACM Conference on Recommender Systems, pp. 169–177 (2019)
13. Song, Q., Cheng, D., Zhou, H.: Towards automated neural interaction discovery for click-through rate prediction. In: 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 945–955 (2020)
14. Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical data: a case study on user response prediction. In: 38th European Conference on IR Research, pp. 45–57. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-30671-1_4
15. Chen, B., Wang, Y., Liu, Z.: Enhancing explicit and implicit feature interactions via information sharing for parallel deep CTR models. In: 30th ACM international Conference on Information & Knowledge Management, pp. 3757–3766 (2021)
16. Cheng, H. T., Koc, L., Harmsen, J.: Wide & deep learning for recommender systems. In: 1st Workshop on Deep Learning for Recommender Systems, pp.7–10 (2016)
17. Guo, H., Tang, R., Ye, Y.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017)

Byzantine-Robust Federated Learning via Server-Side Mixtue of Experts

Xiangyu Fan, Zheyuan Shen, Wei Fan, Keke Yang, and Jing Li(✉)

School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
{ad2018,shenzheyuan,slimfun,ykk}@mail.ustc.edu.cn, lj@ustc.edu.cn

Abstract. Byzantine-robust Federated Learning focuses on mitigating the impact of malicious clients by developing robust algorithms that ensure reliable model updates while preserving privacy. The key insight of the state-of-the-art approaches entails statistical analysis on the local model updates uploaded by clients, concurrently eliminating any malicious updates prior to their aggregation. Some of these methods also require the assistance of server-side data for a reliable root of trust. However, these methods may not perform well and can even disrupt the normal process when the amount of data on the server-side is limited or the structure of the model is complex. We address this challenge by introducing FLSMoE, a novel Byzantine-robust **F**ederated **L**earning approach that utilizes a **S**erver-side **M**ixture **of** **E**xperts. Our approach introduces a novel methodology by implementing a server-side Mixture of Experts(MoE) model, where the model parameters uploaded by individual clients are considered as expert models. Through the utilization of the gating unit within the MoE, even with low server-side data requirement, we are able to effectively identify and exclude malicious clients by assigning appropriate weights to their contributions. Empirically, we show through an extensive experimental evaluation that FLSMoE with low server-side data requirement can effectively mitigate the threat of malicious clients while also exhibiting greater Byzantine-robustness compared to previous Byzantine-robust Federated Learning approaches.

Keywords: Federated Learning · Mixture of Experts · Byzantine-robustness

1 Introduction

In numerous real-world contexts, the exponential growth of internet-connected devices generate an unprecedented volume of private data, creating a challenge for machine learning to efficiently leverage extensive datasets while upholding confidentiality. Federated Learning (FL) [18] represents a prospective distributed learning framework targeted at this objective, focusing on decentralized data,and has garnered considerable attention.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 41–52, 2024.
https://doi.org/10.1007/978-981-99-7022-3_4

However, in light of the distributed architecture inherent in FL, one of major challenges of FL is the potential threat of malicious clients who may attempt to compromise the integrity of the FL process through various attack approaches [15]. Two common types of attacks are data poisoning attacks [3,19], where a malicious client injects poisoned data into its local training dataset, and local model poisoning attacks [1,2,9,23], where a malicious client sends a modified model update to the server. In cases of global model corruption, it has the potential to produce erroneous predictions across a significant portion of testing instances without discrimination [9], or it can be exploited to forecast attacker-selected target labels for particular testing examples, all while retaining its original behavior for other non-target testing instances [1,2,23].

Hence many Byzantine-robust approaches [4–6,12,20,24,25] have been suggested within the FL domain. Most of these approaches rely on statistical analysis of the model updates uploaded by clients to identify and exclude malicious clients or select trustworthy updates. However, some studies [2,9] have shown that even this class of Byzantine-robust Federated Learning approaches can be attacked based on their specific characteristics. To overcome this, Byzantine-robust Federated Learning approaches that require server-side data [5,20] have emerged, which provide a root of trust to the server. However, our subsequent experiments have shown that these approaches do not perform well when there is limited server-side data.

Mixture of Experts (MoE) [17,26] constitutes a category of neural network architecture that combines the predictions of multiple expert models to achieve better overall performance. Each expert model can be trained in a different input subspace, which means that each expert can focus on learning its own specific domain and improve performance as a whole. In FL, MoE has be used to address the heterogeneity of client data [8,13,14,21,27]. Due to potential disparities in data distribution and client features, the overall model's effectiveness might be compromised for specific clients [4]. Through MoE utilization, the diverse data distribution across clients can be aligned with distinct expert models, resulting in improved overall performance. Additionally, MoE can also be used to address the quality differences in client data in FL. Poor-quality clients can be mapped to weaker expert models, while better clients can be mapped to stronger expert models, making the overall model more robust and accurate.

Inspired by this, we propose the server-side MoE approach FLSMoE, where we treat each client's uploaded model parameters as an expert model, train the MoE using a limited volume of data situated on the server side, and use the gating unit of the MoE to assign a weight for each client. Our approach combines the strengths of MoE and FL, providing robustness that significantly reduces the interference of malicious clients. Furthermore, due to MoE's characteristics, our approach can clearly identify and remove poorly performing clients, rather than analyzing updates to speculate like Krum [4], Trimmed Mean [25] or judging based on the direction of model updates like FLTrust [5]. Our approach provides a strong and specific criterion for judgment, making it difficult for malicious clients to deceive the system while also making it difficult to attack our model.

We extensively experiment to evaluate our algorithm's performance and compare against existing methods. We use various datasets and attacks to prove the proposed algorithm achieve higher global test accuracy than FLTrust and others.

In summary, our primary contributions include:

- We introduce FLSMoE, a novel Byzantine-robust Federated Learning approach, which combines robustness and generalization. This represents the first application of MoE on the server-side in Byzantine-robust Federated Learning.
- We address the challenge of low server-side data requirement. Based on our FLSMoE approach, for a server dataset with less than 100 samples, we have achieved accuracy comparable to the FedAvg under normal conditions.
- We thoroughly analyze our approach concerning various existing attacks, while also accounting for the varying sizes of server-side data. And our approach exhibits higher Byzantine-robustness and effectiveness compared to existing Byzantine-robust Federated Learning approaches.

The subsequent sections of the paper are organized as follows. Section 2 offers an overview of related studies on Federated Averaging (FedAvg) algorithm, Byzantine-robust Federated Learning. Section 3 outlines the comprehensive design of the FLSMoE approach. Performance evaluation results and comparisons with other approaches are presented in Sect. 4. Lastly, the paper concludes with Sect. 5.

2 Related Work

2.1 FedAvg [18]

In FedAvg, the model training process involves iterative rounds of local model training on client devices, where each client performs training using its own local data. Subsequently, the central server aggregates client updates via averaging to derive a global model. This iterative and decentralized approach allows the global model to improve over time without requiring the direct exchange of raw data between clients and the server.

2.2 Byzantine-Robust Federated Learning

Based the FedAvg process, it can be inferred that the FedAvg is highly vulnerable and susceptible to malicious clients. Therefore, it is crucial to develop Byzantine-robust Federated Learning.

Krum [4]**.** Krum works by comparing the contributions and similarities among clients to identify reliable clients and exclude suspicious clients that significantly deviate from the consensus. The specific formal expression for the score is as follows:

$$d_i = \sum_{u_j \in S_{i,n-f-2}} ||u_i - u_j||_2^2 \tag{1}$$

In Eq. (1), u_i represents the local model update for client i and $S_{i,n-f-2}$ represents the $n - f - 2$ local model updates that are closest to u_i based on the Euclidean distance. The determination of the global model update involves selecting the local model update from the client with the lowest score, denoting the highest degree of reliability.

Trimmed Mean [25]**.** The trimmed mean is an aggregation rule that computes the average of model parameter values after removing a certain number of extreme values. Specifically, the server sorts the values from all local model updates and discards the largest and smallest values according to a trim parameter, leaving only the middle range of values. The remaining values are then combined through averaging to determine the parameter value in the global model update. This strategy has the capacity to endure a particular count of malicious clients, contingent upon the suitable configuration of the trim parameter.

Median [25]**.** In contrast, the Median aggregation rule also sorts the parameter values but chooses the median value as the parameter for the global model update, instead of using the trimmed mean.

FLTrust [5]**.** FLTrust requires the server-side to have a certain amount of data. While clients undergo training during each iteration, the server independently trains the same model as the client local model for evaluation purposes. FLTrust computes the cosine similarity between the client-uploaded updates and the server-side model update to determine corresponding weights. Additionally, before aggregation, FLTrust trims all client updates to match the size of the server-side update.The formal expression for the global update g is as follows:

$$g = \sum_{i=1}^{m} \frac{ReLU(cos\langle\theta_i, \theta_g\rangle) \cdot \frac{\|\theta_g\|}{\|\theta_i\|} \cdot \theta_i}{\sum_{j=1}^{m} ReLU(cos\langle\theta_j, \theta_g\rangle)} \tag{2}$$

In Eq. (2), θ_i represents the model update for client i, θ_g represents the additional model update on the server-side, $\| \cdot \|$ represents the magnitude of a vector and m represents the total number of clients involved.

However, some studies [9] have shown that Krum, Trimmed Mean and Median are all susceptible to poisoning attacks tailored to their specific characteristics. Furthermore, we believe that relying solely on cosine similarity between parameters, as done in FLTrust, is insufficient for determining client weights and our subsequent experiments have also shown that its performance is poor when the amount of server-side data is limited.

3 Our Approach

3.1 Overview of FLSMoE

Our FLSMoE is expected to overcome previously mentioned challenges and exhibit strong robustness even with limited server-side data, while preserving the normal flow of FL without compromising its functionality in the absence of attacks.

In FLSMoE, the server retains a small portion of clean data, a feasible undertaking, alongside a MoE model consisting of expert models and a gating unit. The overall process is similar to the aforementioned FedAvg [18], but with an additional MoE model trained at the server-side. Specifically, upon receiving the model parameters from the clients, the server proceeds to update the corresponding expert models and freezes their gradients. Subsequently, the MoE model is trained using the server-side data, thereby incorporating the collective knowledge from the experts. After the completion of training, an additional forward pass is performed on the gating unit of the MoE model to obtain the weights assigned to each client.

Intuitively, training a gating unit of a MoE requires a reduced amount of data and is relatively easier compared to training a comprehensive large model. Additionally, the inherent lack of interpretability in machine learning models makes them less susceptible to attacks. Furthermore, the use of MoE allows for assigning appropriate weights to each client, enabling a fair evaluation of their contributions.

3.2 Model of FLSMoE

In FLSMoE, as shown in Fig. 1, the server requires a small amount of clean data and maintains a MoE model $M(\theta_1, \theta_2, \ldots, \theta_n, \beta; x)$ that contains a set of n expert models $F(\theta; x)$ for each client and a gating function $G(\beta; x)$ while θ_k represents the local model parameters for client k, β represents the server gating function parameters, x denotes the input, and n signifies the total count of clients. Every round, after the clients upload their updates, in the MoE model, the expert models $F(\theta; x)$ are frozen and have no gradients, while the gating function is trained by the server. The gating function's output serves as the weight for each client's update during the aggregation process. If the weight is less than a certain threshold, the server will consider the corresponding client as a malicious client and set its weight to 0, thus excluding it from the aggregation in this round.

The formal optimization objective for the server model is given by:

$$argmin_{\beta} L_S(\varSigma_{i=1}^{n}(G(\beta; x)[i])F(\theta_i; x), y) \tag{3}$$

And in (3), n represents the number of clients involved in FLSMoE, L_S represents a task objective of supervised learning, such as cross-entropy loss, while $G(\beta; x)[i]$ represents the output of the gating function for the client i. Finally, the global model parameters θ is aggregated based on the weights generated by the gating function and let the ω_i equal to $G(\beta; x)[i]$ then the global model parameters θ will be formally given by:

$$\theta = \sum_{i=1}^{n} \frac{\omega_i \cdot \theta_i}{\sum_{j=1}^{n} \omega_j} \tag{4}$$

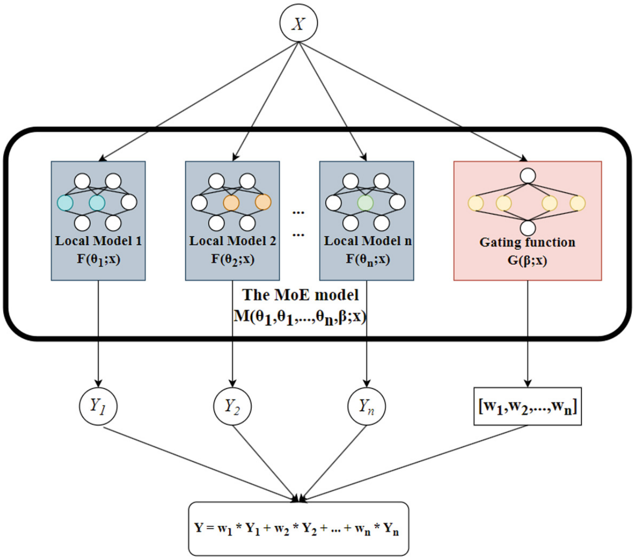


Fig. 1. The MoE model structure in the server. The grey boxes represent no gradient flow, only participating in the forward process, while the pink box represents gradient flow, participating in both the forward and backward processes. (Color figure online)

3.3 FLSMoE Algorithm

In this subsection, we will provide a comprehensive algorithm to make FLSMoE applicable to the majority of FL scenarios. FLSMoE can be broadly categorized into three phases:

Clients Upload Their Updates. Similar to FedAvg [18], each client retrieves the global model from the server, conducts model training using its local dataset, and then uploads its respective update back to the server. And malicious clients may exist among the participating clients, who could potentially poison the aggregation process by uploading maliciously crafted updates. We utilize the function *ClientUpdate* to replace the specific client training procedures.

Server Train the MoE Model with Server-Side Data. Specifically, before aggregation, the server updates the corresponding expert model in the MoE model using the received updates and trains the MoE model with the server-side data. It should be noted that, as mentioned earlier, the expert models in the MoE model do not have gradients, only the gating function has gradients. We utilize the function *ServerMoEUpdate* to replace the server training procedures.

Server Aggregate the Updates. In particular, as demonstrated by Alg.1, after training, the MoE model is forwarded with the server-side data to acquire the current gating function's output, which allocates a weight to each client. Then the server aggregates the updates based on the weights assigned by the gating function, resulting in a global model, rather than basing the aggregation on the size of the clients' datasets.

Algorithm 1. Aggregate

```
Require: server MoE model structure $M$ including gating function $G$; server gating
    function parameters $\beta$; server-side dataset $D_{server}$; the set of clients participating
    in the aggregation $S_t$; threshold $\epsilon$; clients' local model parameters $\theta_i$, $i = 1, 2, \ldots, n$.
Ensure: global model parameters $\theta$
// Assign weights
for batch $(x, y) \subset D_server$ do
    // $W$ is a dict $\{client\ id : L\}$ for all clients, the value is a list.
    // Each batch, the gating function outputs a weight list $V$.
    // The length of $V$ is the overall count of clients throughout the procedure.
    // Each element of $V$ appends the corresponding $L$ in $W$.
    $W \leftarrow G(\beta; x)$
end for
for $i$ in $W$ do
    $W[i] \leftarrow mean(W[i])$
end for
// Aggregate the local model updates.
total_weight $= \sum_{i \in S_t} ReLU(W[i] - \epsilon)$
$\theta \leftarrow \frac{\sum_{i \in S_t} ReLU(W[i]-\epsilon) \cdot \theta_i}{total\_weight}$
return $\theta$
```

In conclusion, we have derived a comprehensive algorithm, as illustrated by Algorithm 2, that encompasses all the necessary steps and considerations discussed earlier.

4 Experiments

4.1 Settings

The experiments in this paper were carried out utilizing **PyTorch** and the model computations were performed on a **GTX 1080 Ti**. We use two datasets, five baselines, and four types of attacks to demonstrate the robustness and efficiency.

The following will describe the experimental settings. For each scenario, we executed the experiment ten times and computed the average to obtain the final result. Moreover, we found that their variances were so small that they could be ignored.

Datasets. We used **MNIST** [7] and **CIFAR10** [16] as the experimental datasets. Both of them are image classification datasets.

Algorithm 2. Byzantine-robust Federated Learning via Server-side Mixture of Experts with Low Data Requirement

```
Require: local model structure $F$; server MoE model structure $M$ including gating
    function $G$; server gating function parameters $\beta$; global communication rounds $T_g$;
    server training epochs $E_s$; client training epochs $E_c$; server learning rate $\eta_s$; client
    learning rate $\eta_c$; $n$ clients with local training datasets $D_i$, $i = 1, 2, \ldots, n$; randomly
    initialized client local model parameters $\theta_i$, $i = 1, 2, \ldots, n$; server-side dataset $D_s$;
    number of clients sampled per round $\tau$; client loss function $L_c$; server loss function
    $L_s$; threshold $\epsilon$.
Ensure: global model $\theta$
// Phase 0: initialization
for each client do
    send its local model parameters $\theta_i$ to server
end for
server receives the client local model parameters, creates corresponding expert
    models, and initializes the gating function
for $t$ in $T_g$ do
    $S_t \leftarrow$ randomly select $\tau$ out of $n$ clients
    // Phase 1: Clients Update
    for each client $i$ in $S_t$ in parallel do
        $\theta_i \leftarrow \mathit{ClientUpdate}(F, E_c, \eta_c, D_i, \theta, L_c)$
        send its local model parameters $\theta_i$ to server
    end for
    server receives the local model parameters and updates corresponding experts
    // Phase 2: Server MoE Update
    $\beta \leftarrow \mathit{ServerMoEUpdate}(M, G, \beta, E_s, \eta_s, D_s, L_s, \theta_1, \theta_2, \ldots, \theta_n)$
    // Phase 3: Assign weights for each participant and aggregate
    $\theta \leftarrow \mathit{Aggregate}(M, G, \beta, D_s, S_t, \epsilon, \theta_1, \theta_2, \ldots, \theta_n)$
end for
```

Baselines. We compared approach with **FedAvg with no defense mechanism** [18] against Byzantine attacks, and with other Byzantine-robust approaches, including **Krum** [4], **Trimmed Mean** [25], **Median** [25] and **FLTrust** [5], to demonstrate the superiority of our approach in defensing attacks.

Attacks. To demonstrate the robustness of our approach, we assume that attackers have full knowledge. We implement one data poisoning attack: **Label-flip Attack** [22]. In addition, we conduct three types of local model poisoning attacks: **Krum Attack** [9], **Trimmed Mean Attack** [9] (also called Median Attack), and **Omniscient Attack**. The Omniscient Attack is designed by us, where the attacker always uploads the negation of the global model parameters.

Model Architecture and Parameter Settings. We utilize CNN [11] to implement the global model and local model, and realize the gating function of the MoE model with MLP [10]. We set up a total of 100 clients, out of which 30 are malicious and the server selects 100 clients in each round. The server dataset is fixed to have ten samples for each class, to demonstrate that our approach only requires a small amount of data.

4.2 Results

The performance of the proposed FLSMoE approach is assessed and contrasted with state-of-the-art Byzantine-robust Federated Learning methods under diverse attack scenarios. And results are shown in Table 1 and Table 2.

Table 1. The mean global test accuracy values across five baselines and FLSMoE on **CIFAR10**. Bold font signifies the highest accuracy among all approaches.

	FedAvg	Krum	Trimmed Mean	Median	FLTrust	FLSMoE
No Attack	0.8255	0.7504	**0.8295**	0.818	0.7961	0.8218
Label-flip Attack	0.6358	0.4526	0.6212	0.6489	0.6332	**0.6646**
Krum Attack	0.8025	0.6742	0.8098	0.8097	0.7638	**0.8158**
Trimmed Mean Attack	0.1	0.7336	0.741	0.7328	0.7655	**0.7963**
Omniscient Attack	0.1	0.7418	0.6138	0.6054	**0.798**	0.7785

Table 2. The mean global test accuracy values across five baselines and FLSMoE on **MNIST**. Bold font signifies the highest accuracy among all approaches.

	FedAvg	Krum	Trimmed Mean	Median	FLTrust	FLSMoE
No Attack	**0.991**	0.9876	0.9894	0.9896	0.9907	0.9899
Label-flip Attack	0.9327	0.8919	0.9483	0.9455	0.735	**0.9724**
Krum Attack	0.9889	0.1459	0.9878	0.9894	0.9889	**0.9906**
Trimmed Mean Attack	0.8395	0.9861	0.8771	0.9686	0.9893	**0.99**
Omniscient Attack	0.1135	0.9854	0.9623	0.9829	0.9891	**0.9902**

Our Approach is Effective and Better than Others. Experimental findings illustrate the enhanced roubstness of our approach in comparison to alternative Byzantine-robust Federated Learning strategies, showcasing superior performance across most scenarios. In the only case where our approach is outperformed, we still achieve a similar performance to the best-performing approach. Specifically, as shown in Fig. 2(a) on the CIFAR10 dataset, under omniscient attack, FLTrust achieves a test accuracy of 0.79, which is the best result, while our approach achieves a similar performance with a test accuracy of 0.77, which is the second best. And to demonstrate the robustness of our approach, we present a typical scenario which is on the MNIST dataset under Label-flip Attack in Fig. 2(b).

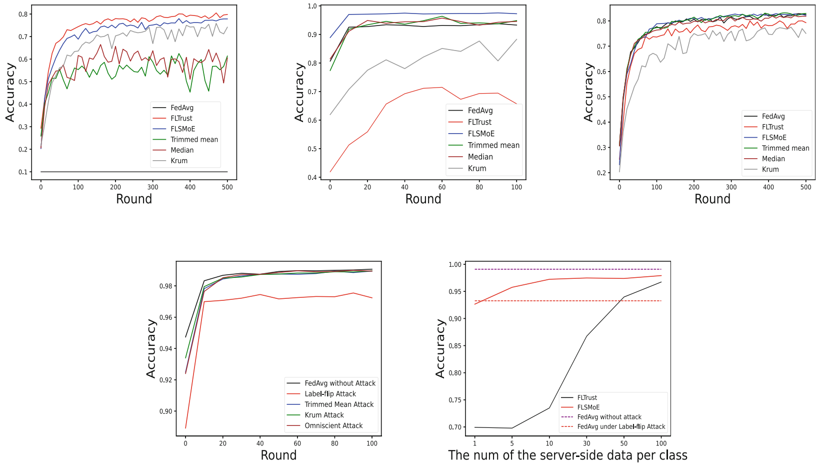


Fig. 2. The global test accuracy vs. the global communication rounds for (a) FLSMoE and five baselines on the CIFAR10 under Omniscient Attack; (b) FLSMoE and five baselines on the MNIST under Label-flip Attack; (c) FLSMoE and five baselines on the CIFAR10 under normal conditions; (d) FedAvg without attack and FLSMoE under different attacks on the MNIST. (e) The global test accuracy vs. the number of the server-side data per class for on the MNIST for FedAvg without attack and under Label-flip Attack; FLSMoE and FLTrust under Label-flip Attack.

Our Approach Does Not Interfere with the Regular FedAvg Process. Without attack, our approach does not affect the normal execution of the process and achieves similar performance to FedAvg. Despite the presence of malicious clients, our approach can still achieve a performance similar to the performance of FedAvg when not subjected to attacks in certain instances. And it's important to note that our approach and FedAvg converge at the same speed, because we did not add any extra tasks to the client-side and training the gating units on the server-side incurs negligible computational overhead for a powerful server. As an example, on the CIFAR10 dataset, under normal conditions, as shown in Fig. 2(c), the proposed approach shows comparable results with FedAvg, achieving a test accuracy of 0.82, while Krum and FLTrust achieve a test accuracy of 0.75 and 0.79 respectively. Moreover, on the MNIST dataset, under different attack scenarios, also shown in Fig. 2(d), FLSMoE also achieves similar testing accuracy rates to that of FedAvg without attacks.

Our Approach Requires Minimal Server-Side Data. To demonstrate that our approach achieves superior accuracy with fewer server-side data, we conducted additional experiments to compare it with FLTrust. As shown in Fig. 2(e), we maintained all other conditions unchanged and only modified the server-side data to evaluate the final test accuracy of FLSMoE and FLTrust under the Label-flip attack on the MNIST dataset. Additionally, for ease of comparison, we also included the final test accuracy of FedAvg under normal conditions and the Label-flip attack in the same figure. It can be observed that FLSMoE outperforms the attacked FedAvg with only 5 data samples per class, while FLTrust requires 50 data samples per class to surpass its performance.

5 Conclusion

In this paper, we present FLSMoE, a novel FL framework for countering Byzantine attacks. Our approach employs a server-side MoE, ensuring robustness and preserving data privacy. Extensive empirical evaluations confirm the superior efficacy of FLSMoE compared to other Byzantine-robust Federated Learning approaches. Remarkably, FLSMoE exhibits exceptional resilience against a substantial number of malicious clients, even in data-limited scenarios.

We aim to evaluate FLSMoE practically through smartphone deployment. Our focus is on designing a streamlined parallel interface for multi-CPU servers to minimize computation time. Additionally, we plan to explore asynchronous user participation during training as part of our future research.

References

1. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics, pp. 2938–2948. PMLR (2020)
2. Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S.: Analyzing federated learning through an adversarial lens. In: International Conference on Machine Learning, pp. 634–643. PMLR (2019)
3. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
4. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
5. Cao, X., Fang, M., Liu, J., Gong, N.Z.: FLTrust: byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv:2012.13995 (2020)
6. Chen, Y., Su, L., Xu, J.: Distributed statistical machine learning in adversarial settings: byzantine gradient descent. Proc. ACM Meas. Anal. Comput. Syst. **1**(2), 1–25 (2017)
7. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)
8. Deng, Y., Kamani, M.M., Mahdavi, M.: Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461 (2020)
9. Fang, M., Cao, X., Jia, J., Gong, N.Z.: Local model poisoning attacks to byzantine-robust federated learning. In: Proceedings of the 29th USENIX Conference on Security Symposium, pp. 1623–1640 (2020)
10. Gardner, M.W., Dorling, S.: Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences. Atmos. Environ. **32**(14–15), 2627–2636 (1998)
11. Gu, J., et al.: Recent advances in convolutional neural networks. Pattern Recogn. **77**, 354–377 (2018)

12. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning, pp. 3521–3530. PMLR (2018)
13. Guo, B., Mei, Y., Xiao, D., Wu, W.: PFL-MoE: personalized federated learning based on mixture of experts. In: U, L.H., Spaniol, M., Sakurai, Y., Chen, J. (eds.) APWeb-WAIM 2021, Part I. LNCS, vol. 12858, pp. 480–486. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85896-4_37
14. Isaksson, M., Zec, E.L., Cöster, R., Gillblad, D., Girdzijauskas, Š.: Adaptive expert models for personalization in federated learning. arXiv preprint arXiv:2206.07832 (2022)
15. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends® Mach. Learn. **14**(1–2), 1–210 (2021)
16. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report 0, University of Toronto, Toronto, Ontario (2009)
17. Masoudnia, S., Ebrahimpour, R.: Mixture of experts: a literature survey. Artif. Intell. Rev. **42**(2), 275 (2014)
18. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
19. Nelson, B., Barreno, M., Chi, F.J., Joseph, A.D., Rubinstein, B.I., Saini, U., Sutton, C., Tygar, J.D., Xia, K.: Exploiting machine learning to subvert your spam filter. LEET **8**(1–9), 16–17 (2008)
20. Parsaeefard, S., Etesami, S.E., Garcia, A.L.: Robust federated learning by mixture of experts. arXiv preprint arXiv:2104.11700 (2021)
21. Reisser, M., Louizos, C., Gavves, E., Welling, M.: Federated mixture of experts. arXiv preprint arXiv:2107.06724 (2021)
22. Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L.: Data poisoning attacks against federated learning systems. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020, Part I. LNCS, vol. 12308, pp. 480–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58951-6_24
23. Xie, C., Huang, K., Chen, P.Y., Li, B.: DBA: distributed backdoor attacks against federated learning. In: International Conference on Learning Representations (2020)
24. Yang, H., Zhang, X., Fang, M., Liu, J.: Byzantine-resilient stochastic gradient descent for distributed learning: a Lipschitz-inspired coordinate-wise median approach. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 5832–5837. IEEE (2019)
25. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659. PMLR (2018)
26. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE Trans. Neural Netw. Learn. Syst. **23**(8), 1177–1193 (2012)
27. Zec, E.L., Mogren, O., Martinsson, J., Sütfeld, L.R., Gillblad, D.: Specialized federated learning using a mixture of experts. arXiv preprint arXiv:2010.02056 (2020)

CDAN: Cost Dependent Deep Abstention Network

Bhavya Kalra and Naresh Manwani(✉)

Machine Learning Lab, KCIS, IIIT Hyderabad, Hyderabad, India
naresh.manwani@iiit.ac.in

Abstract. This paper proposes deep architectures for learning instance-specific abstain (reject) option multiclass classifiers. The proposed approach uses novel bounded multiclass abstention loss for multiclass classification as a performance measure. This approach uses rejection cost as the rejection parameter in contrast to coverage-based approaches. To show the effectiveness of the proposed approach, we experiment with several real-world datasets and compare them with state-of-the-art coverage-based and cost-of-rejection-based techniques. The experimental results show that the proposed method improves performance over the state-of-the-art approaches.

Keywords: Reject Option · Multiclass Classification · Deep Learning

1 Introduction

In many classification problems, the cost of misclassification is very high (e.g., healthcare, financial decision, etc.). In such cases, it is more appropriate to avoid (reject) the decision-making on confusing examples, especially when the cost of the rejection option is much lesser than the cost of misclassification. The reject option is to refrain from making a classification decision on some samples. Classifiers having such an option are called reject option classifiers. These classifiers can be vital when learning in critical tasks such as medical diagnosis [17], speech emotion recognition [32], text categorization [10], software defect prediction [23] [6], financial forecasting [29], genomics [14], crowd-sourcing [20], social discrimination control [16], safety in autonomous vehicles [24] etc. The availability of the reject option improves the classifier's reliability in such decision support systems.

The desired goal of reject option classification is to minimize the risk by achieving high accuracy on most samples while minimizing the number of rejections. There have been two major rejection options classification approaches: (a) coverage-based and (b) rejection cost-based. Coverage-based methods [26] don't assume any cost of rejection. In such techniques, two metrics evaluate the model's performance: (1) selective risk, defined as the misclassification rate computed over examples accepted for prediction, and (2) coverage corresponding to the fractions of examples accepted for prediction. An optimal strategy for the bounded-improvement model [8] maximizes the coverage because the selective

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 53–65, 2024.
https://doi.org/10.1007/978-981-99-7022-3_5

risk does not exceed a target value. On the other hand, cost-based approaches assume a cost involved for every rejection (pre-decided). The goal is to minimize the number of rejections and misclassifications on unrejected examples. Below we discuss the existing approaches for reject options and classify them into two categories: (a) kernel-based and (b) neural network based.

1.1 Kernel Based Approaches for Learning with Abstention

Abstaining classifiers have been explored extensively in binary classification settings. Generalized hinge SVM [1], double hinge SVM [13], double ramp SVM [22], SDR-SVM [30], max-hinge SVM and plus-hinge SVM [4] etc. are some variants of support vector machine (SVM) for abstaining classifiers. Nonlinear classifiers in these approaches are learned using kernel functions. A boosting algorithm for abstaining classifiers is proposed in [3]. Active learning of abstaining classifiers is discussed in [31].

Various algorithms have been proposed in the multiclass setting, which extends naturally to one vs. all implementations. In [27], excess risk bounds of Crammer-Singer loss [5] and one vs. all hinge surrogates [28] are established. They propose a new surrogate, a binary encoded prediction method, and excess risk bounds. These algorithms also require a threshold hyper-parameter for the risk coverage trade-off. These approaches face two significant challenges. (a) The approach to relying on kernel tricks to learn nonlinear classifiers makes them infeasible for big data. (b) Parameter ρ, which captures rejection bandwidth, is considered independent of the instances (i.e., $\rho(\mathbf{x}) = \rho,\ \forall \mathbf{x} \in \mathcal{X}$).

1.2 Neural Network Based Approaches for Learning with Abstention

A post-processing-based deep learning approach for reject options has been explored in [11] where best-abstaining thresholds are found for each class using the softmax function for a pre-trained network. Recently, coverage-based methods have been proposed for learning with abstention [8,12]. Coverage is defined as the ratio of samples that the model does not reject. Such approaches do not take the cost of rejection d as an input. The purpose of the selective function is to select enough examples to match the coverage condition. Such methods try to learn an appropriate selection function and a classification function in a deep learning setting for a given coverage. Learning is based on optimizing risk-coverage trade-offs. As this approach does not consider rejection cost d in their objective function, it can avoid rejecting hazardous examples. This, in particular, becomes a severe issue in high-risk situations (e.g., healthcare systems, etc.). The work in [33] assumes the reject option as another category and extends the cross-entropy for the same. The modified cross-entropy function uses a learnable hyperparameter expressing the degree of penalty for abstaining from a sample. However, considering the rejection region as a different class itself may not serve the fundamental purpose of rejection as it captures the gray regions in the feature space. In [15], the authors propose a cost-based rejection deep neural network, which works only for binary classification tasks.

Learning to defer is a slightly related problem to learning abstaining classifiers. Learning to defer uses abstaining classifiers with a human expert in the loop. Abstaining classifiers discussed so far are trained independently from the human decision-maker; however, when a human expert and an abstaining classifier work together, rejection parameters must be learned adaptively. For binary classification, [21] propose learning to defer model. In a multiclass setting, such models are presented in [25].

1.3 Proposed Approach

This paper introduces deep neural network architectures for multiclass classification with abstention, which involves cost for rejection also. This paper presents a novel loss function that supports multiclass classification with abstention. This is the first paper that explores the abstain option in multiclass deep neural networks (DNN), where the cost of rejection can be utilized as a hyperparameter to train the network. Note that the proposed approach does not consider abstention as a separate class.

Key Contributions: Our key contributions in this paper are as follows.

1. We propose a new loss function for the multiclass abstention option classifier, called *bounded multiclass abstention* (L_d^{BMA}) loss.
2. We propose a novel deep abstain network called CDAN. We consider two variants of CDAN with (a) input-independent rejection function and (b) input-dependent rejection function.
3. We show the effectiveness of the proposed approach by comparing its performance with state-of-the-art algorithms on benchmark datasets.

2 Multiclass Reject Option Classifier

Let $\mathcal{X} \subseteq \mathbb{R}^D$ be the feature space and $\mathcal{Y} = [k]$ be the label space, where $[k] = \{1, \ldots, k\}$. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be the training dataset such that $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. For a given cost of rejection d, the objective here is to learn a set of decision functions $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_k(\mathbf{x})] \in \mathbb{R}^k$ and corresponding rejection bandwidth parameters $\boldsymbol{\rho}(.) = [\rho_1(.), \rho_2(.), \ldots, \rho_k(.)] \in \mathbb{R}^k$. Here $\boldsymbol{\rho}$ can be dependent on feature vector $\mathbf{x}$. Let $\hat{y} = \arg\max_{r \in [k]} h_r(\mathbf{x})$, then the multiclass reject option classifier $f : \mathcal{X} \rightarrow [k] \cup \{\text{reject}\}$ is given as follows.

$$f(\mathbf{h}(\mathbf{x}), \boldsymbol{\rho}) = \begin{cases} \hat{y}, & h_{\hat{y}}(\mathbf{x}) - \max\limits_{y' \neq \hat{y}} h_{y'}(\mathbf{x}) > \rho_{\hat{y}} \\ \text{reject}, & h_{\hat{y}}(\mathbf{x}) - \max\limits_{y' \neq \hat{y}} h_{y'}(\mathbf{x}) \leq \rho_{\hat{y}} \end{cases} \tag{1}$$

Note that here we allow different rejection bandwidths for different classes. To measure the quality of the prediction, we use $0-d-1$ loss (denoted as L_d)

described below.

$$L_d(f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}),y) = \begin{cases} 1 & f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}) \neq y \text{ and } f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}) \neq \textit{reject} \\ d & f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}) = \textit{reject} \\ 0 & f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}) = y \end{cases} \tag{2}$$

Cost of rejection (d) can take values in the range $[0, \frac{k-1}{k}]$ [27]. For a given loss L, score functions $\mathbf{h}$ and rejection bandwidth parameters $\boldsymbol{\rho}$ and cost of rejection d, risk is written as $R_d^L(\mathbf{h},\boldsymbol{\rho}) = \mathbb{E}_{\mathcal{X}\times\mathcal{Y}}\left[L_d(f(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho}),y)\right]$. For the multiclass reject option classifier, the risk under L_d loss is minimized by the generalized Bayes classifier f_d^* [2,27], which is described as follows.

$$f_d^*(\mathbf{x}) = \begin{cases} \arg\max_{r\in[k]} p_{\mathbf{x}}(r) & \max_{r\in[k]} p_{\mathbf{x}}(r) \geq 1-d \\ \textit{reject} & \textit{else} \end{cases} \tag{3}$$

where $p_{\mathbf{x}}(r) = P(Y=r|X=\mathbf{x})$. Thus, f_d^* is composed of $\mathbf{h}^*$ and $\boldsymbol{\rho}^*$ as follows.

$$\begin{aligned} h_i^*(\mathbf{x}) &= P(Y=i|\mathbf{x}), && \forall i \in \{1,\ldots,k\} \\ \rho_i^*(\mathbf{x}) &= 1 - d - \max_{j\neq\hat{y}} P(Y=j|\mathbf{x}), && \forall i \in \{1,\ldots,k\} \end{aligned}$$

where $\hat{y} = \arg\max_{l\in\{1,\ldots,k\}} P(Y=l|\mathbf{x})$. Minimizing empirical risk under loss L_d is computationally difficult as L_d is not continuous. In practice, surrogate loss functions, which are continuous upper bound to L_d, are used to learn the classifiers.

3 Bounded Multiclass Abstention Loss

We first describe a new loss function for multiclass reject option classification called bounded multiclass abstention (BMA) loss. The BMA loss L^{BMA} is an extension of the double ramp loss [22] for multiclass reject option problems. L^{BMA} loss differs from double ramp loss in two aspects. (a) L^{BMA} loss is designed for multiclass cases, whereas double ramp loss is designed for the binary case. (b) L^{BMA} loss also accommodates different rejection bandwidths corresponding to different classes. Rejection bandwidths can be instance specific also. On the other hand, double ramp loss considers the same rejection bandwidth parameter, which is a scalar.

The intuition behind the construction of loss L^{BMA} is as follows. Loss L_d can be written as a sum of two-step functions (one has a step at ρ and the other and step at $-\rho$.). Each of the step functions is approximated using a ramp function (which is continuous). Summing these two ramps will provide L^{BMA} loss. Given $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}),\ \ldots\ ,h_k(\mathbf{x})] \in \mathbb{R}^k$ and $\boldsymbol{\rho} = [\rho_1,\ \ldots,\ \rho_k] \in \mathbb{R}^k$, we define L^{BMA} loss as follows.

$$\begin{aligned} L^{\text{BMA}}(\mathbf{h}(\mathbf{x}),\boldsymbol{\rho},y) = &\frac{d}{\mu}\Big[[\mu - t + \rho_y]_+ - [-\mu^2 - t + \rho_y]_+\Big] \\ &+ \frac{1-d}{\mu}\Big[[\mu - t - \rho_y]_+ - [-\mu^2 - t - \rho_y]_+\Big] \end{aligned} \tag{4}$$

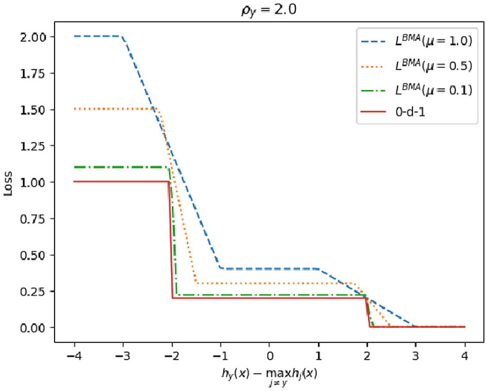


Fig. 1. Bounded multiclass abstention loss for different values of μ.

where $t = h_y(\mathbf{x}) - \max_{j \neq y} h_j(\mathbf{x})$, $\mu > 0$ and $[a]_+ = \max(0, a)$. L^{BMA} is a continuous upper bound for L_d (Eq. (2)), which can be easily seen from Theorem 1 of [22]. Figure 1 shows BMA loss for $\rho_y = 2$. The rejection region for L_{BMA} is between $[-\rho_y, \rho_y]$. Loss L^{BMA} can achieve zero value when $h_y(\mathbf{x}) - \max_{j \neq y}\ h_j(\mathbf{x}) \geq \rho_y + \mu$. The risk under L^{BMA} is given as,

$$R_{BMA}(\mathbf{h}, \boldsymbol{\rho}) = \mathbb{E}_{\mathcal{X} \times \mathcal{Y}}[L_{BMA}(\mathbf{h}(\mathbf{x}), \boldsymbol{\rho}, y)].$$

In this paper, we model $\mathbf{h}(.)$ and $\boldsymbol{\rho}$ using a neural network and learn the parameters by minimizing the risk R_{BMA} described above.

4 Proposed Approach: Cost Dependent Abstention Network (CDAN)

This section proposes a new deep-learning approach for multiclass classification with a reject option. This approach uses the proposed loss function L_d^{BMA}. Since the proposed approach depends on the cost of rejection, it is called *cost-dependent abstention network (CDAN)*. The proposed model integrates the decision functions $\mathbf{h}(.) = [h_1(.), h_2(.), \ldots, h_k(.)]$ and the rejection functions $\boldsymbol{\rho}(.) = [\rho_1(.), \rho_2(.), \ldots, \rho_k(.)]$ into a single DNN model. The main body of the CDAN can have fully connected, convolution, or recurrent layers, depending on the problem at hand. We propose two different architectures for CDAN as follows. (a) CDAN Input Independent Rejection: in this architecture, the rejection bandwidth parameter vector $\boldsymbol{\rho}$ is assumed to be independent of the input vectors, i.e., $\boldsymbol{\rho}(\mathbf{x}) = \boldsymbol{\rho}$, $\forall \mathbf{x}$. (b) CDAN Input Dependent Rejection: here, the rejection bandwidth parameter vector $\boldsymbol{\rho}$ depends on the input vector. In both variants, the objective is to optimize the decision function ($\mathbf{h}(\mathbf{x})$) and rejection bandwidths ($\boldsymbol{\rho}(\mathbf{x})$). The utility of each of the variants depends on the dataset's size and the dataset's type.

4.1 CDAN: Input Independent Rejection

Here, we assume that the rejection bandwidth is independent of the input feature vector. The architecture of input-independent rejection CDAN is given in Fig. 2. In this setting, the model assumes that $\boldsymbol{\rho}(\mathbf{x}) = \boldsymbol{\rho}$, $\forall \mathbf{x} \in \mathcal{X}$. CDAN for input independent rejection has $2k$ output nodes, k nodes for predicting scores for each class ($\mathbf{h}(\mathbf{x})$), and k nodes outputting values of rejection bandwidth parameters ($\boldsymbol{\rho}$). The input data $\mathbf{x}$ is fed into the fully connected (FC) layers, while k neurons feed constant values into the rejection heads. The associated weight to rejection heads then becomes the rejection parameter. The task of the k prediction heads is to learn the k decision surfaces $\mathbf{h}(\mathbf{x})$, and the rejection heads are to learn the corresponding k rejection bandwidths. The network parameters are learned using the backpropagation algorithm, which minimizes L_d^{BMA}.

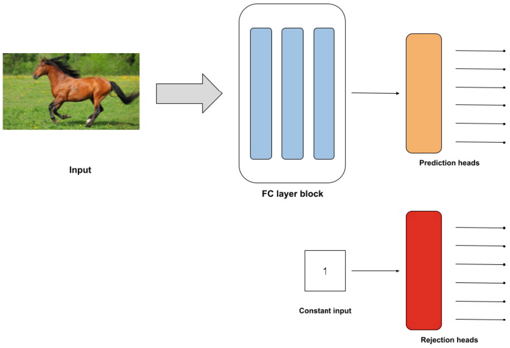


Fig. 2. Input Independent Rejection Architecture for CDAN

4.2 CDAN: Input Dependant Rejection

The input-dependent CDAN (CDAN-dependent) is an abstention model where rejection bandwidths depend on the specific instance (Fig. 3 and Fig. 4). The rejection heads are fed input from the final layer of the main body block. Thus, the rejection head outputs become a function of the instance. These outputs are refined and assist in finding optimal rejection bandwidths, unlike constant inputs, when the variance and dimensionality of data are high. We add an auxiliary head, Fig. 4 for larger datasets. The core architecture has two sets of output heads, k prediction heads ($\mathbf{h}(\mathbf{x})$) and k rejection heads ($\boldsymbol{\rho}(\mathbf{x})$). The depth and size of a fully connected network preceding these two heads are independent and vary depending on the underlying task.

4.3 CDAN: Input Dependent Rejection with Auxiliary Heads

In this architecture, we have an additional set of k auxiliary heads ($\mathbf{g}(\mathbf{x})$) as shown in Fig. 4. We compute cross-entropy loss using $\mathbf{g}(\mathbf{x})$ to the actual class

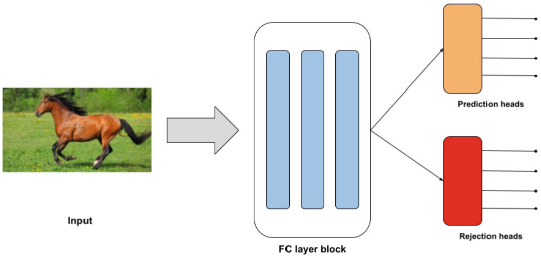


Fig. 3. Input Dependent CDAN Architecture without Auxiliary head

labels. Thus, Input Dependent CDAN aims to find optimal decision functions and rejection bandwidth parameters in an auxiliary task (simple classification task without rejection). A similar idea has been used in [12]. Therefore, the overall objective for CDAN is to minimize a weighted combination of cross-entropy loss L^{CE} and BMA loss L_d^{BMA} as follows.

$$L_{cdan} = \alpha L_d^{\mathrm{BMA}} + (1-\alpha)L^{CE} \tag{5}$$

Here, α is a hyper-parameter chosen in the interval (0, 1). We observe that during the initial learning phase, the auxiliary heads play an important role in assimilating complex features from the main body block. Thus, it leads to more meaningful representations in the shared layer. This facilitates the prediction and rejection heads to build better features and optimize themselves to minimize the overall loss. Without the auxiliary loss, the network has two ways to reduce the loss. It can either opt to reject everything or focus on improving accuracy. Depending on the initialization parameters and cost of rejection, it opts for one of those options before accurate low-level features are assembled. With the addition of auxiliary loss, the network can focus on learning optimal features for prediction and rejection parameters instead of solely on building features only to improve accuracy.

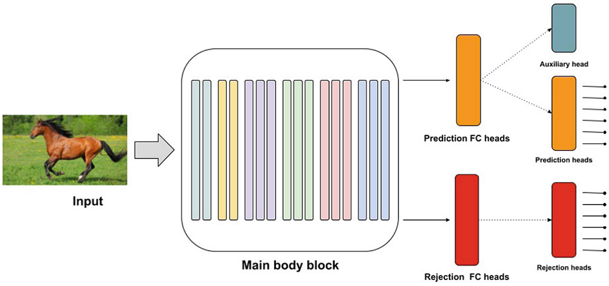


Fig. 4. Input Dependent CDAN with an Auxiliary head

Remarks: CDAN-Independent is a simpler model than CDAN-Dependent. It depends on the application which method is more suitable. In the applications where overlap between classes (region of confusion) is uniform across the feature space, CDAN-Independent is sufficient. On the other hand, when the overlap between the classes is nonuniform across the feature space, CDAN-Dependent is more beneficial.

5 Experiments

To show the efficiency of the proposed approach, we perform experiments on various benchmark datasets. We compare the proposed approach with various state-of-the-art baseline algorithms. The complete details are as follows.

5.1 Datasets Used

We use two kinds of datasets for our experiments: small and large. Some baseline approaches (e.g., kernel-based methods) are more optimized for smaller datasets and fail to converge for larger ones. On the other hand, deep learning based approaches can handle even larger datasets. The two kinds of datasets used are described below.

- Small Datasets: Image, Satmage and covertype [7]. We use these datasets to experiment with kernel-based approaches and non-deep methods.
- Image Datasets: SVHN [9], CIFAR-10 [18], Fashion MNIST [19]. We used these datasets to experiment with deep learning based approaches.

5.2 Baselines for Large Datasets Experiments:

We use the following baselines methods for CIFAR-10, SVHN, and Fashion MNIST.

- SelectiveNet (SNN) [12]: a deep neural architecture with an integrated reject option that optimizes a prediction and selection function. Selective Net (SNN) takes as input a coverage parameter. Coverage denotes the fraction of points that are not abstained by the network.
- DAC: Deep abstaining classifier, a deep neural network trained with a modified cross-entropy loss function introduced in [33] to accommodate an abstain option. DAC takes the abstention rate as an input parameter.
- Softmax Response (SR): The SR method [11], is a post-processing method that makes a selective prediction on samples with a maximum softmax confidence score above a certain threshold based on a pre-trained network.

5.3 Baselines for Small Datasets Experiments:

We use the following baseline methods for small datasets.

- BEP: multiclass reject option classifier introduced in [27] which minimizes the double hinge loss.
- OvA: multiclass reject option classifier based on Hinge loss also proposed in [27].
- Softmax Response (SR) [11].
- SelectiveNet(SNN) [12]: We remove the auxiliary network from SNN for these experiments.
- DAC: Deep abstaining classifier [33].

5.4 Experimental Settings

For Image and Satimage datasets [7] datasets, we perform ten repetitions of ten-fold cross-validation for each of the baseline methods and the proposed approach. We perform ten repetitions of five-fold cross-validation for Covertype, SVHN, CIFAR-10, and Fashion MNIST datasets. We do these for the cost of rejection (d) varying from $[0.1, \frac{k-1}{k}]$ with a step size of 0.1. We monitor the accuracy (on unrejected samples) vs. rejection rate curves for various values of d.

5.5 Network Architecture and Implementation Details

We use the sample independent (Fig. 2) and sample dependent (Fig. 3) architectures with relu activation at each layer, including the final layer at the rejection head. However, no activation function is applied to the final outputs at the prediction head.

Architecture for Small Datasets: For small datasets, the network contains three layers, each of 128 neurons, and trains it with a learning rate scheduler beginning with an initial learning rate of $1e-2$. Batch normalization layers follow each layer. We use SGD optimizer with gradient clipping to train the networks. The loss function parameter μ is set to 2. For each run, we train the network for 150 epochs.

Architecture of Large Datasets: For large datasets, the architecture follows the VGG-net architecture, but instead of 3 fully connected layers of size 1024, we use a single layer of size 512. We also deploy batch normalization and dropout layers. For our network, the final layer of the rejection head uses a sigmoid as the activation function, which bounds values of rejection parameters between 0 and 1. This reduces the prediction head function space, helps the network learn appropriate head functions, and leads to stable models. When rejection bandwidth parameters aren't bounded in $[0, 1]$, the final models have either shallow rejection values or very high rejection values leading to high variance in the corresponding accuracy results. We run all the experiments for 250 epochs and initialize the learning rate scheduler with an initial learning rate of $1e - 2$

drops by 0.5 when the validation loss stops changing by $1e-4$. The batch size used for all the experiments is set to 128. The μ parameter gives optimal results at $\mu=1$. We also do image data augmentation with width and height parameter range set to 0.1. We also allow horizontal image flips and a rotation range of 15. The α parameter is also set to 0.5

As our approach is a cost-based method, rejection cost dictates the coverage. To get a wide range of rejection rates (or coverage), we select the cost of rejection d parameter for our CDAN methods from $\{0.0001, \frac{k-1}{k}\}$. We vary the abstention rate parameter for DAC [33] from [0.1, 1.0] with a step size of 0.1. Similarly, the coverage parameter from SNN [12] is varied from [0.1, 1.0] with a step size of 0.1. We plot the rejection rate vs. accuracy for each algorithm to compare the various abstention methods (Figs. 5 and 6).

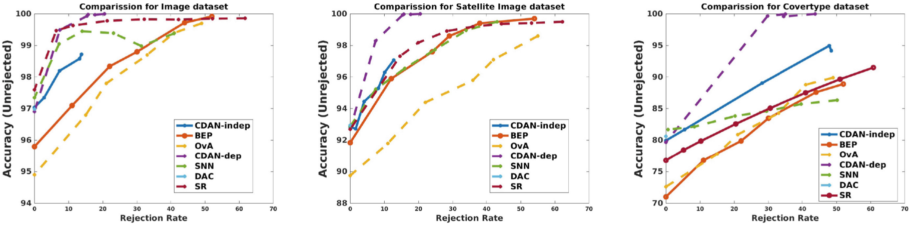
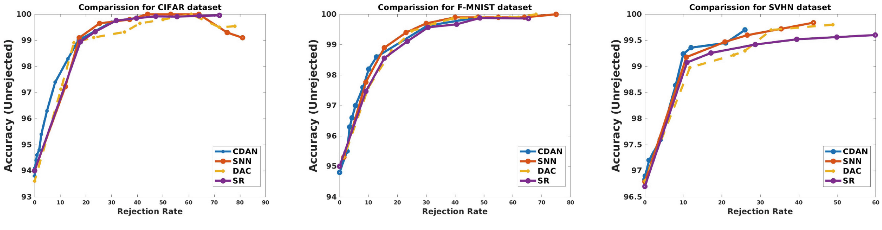


Image Dataset SatImage Dataset Covertype Dataset

Fig. 5. Results on Small Datasets



CIFAR-10 Dataset Fashion MNIST Dataset SVHN Dataset

Fig. 6. Results on Large Dataset

5.6 Empirical Observations on Small Datasets

We evaluated the effectiveness of CDAN by plotting accuracy and rejection rates for all the methods. We observed that CDAN-independent does much better, 4–5%, on average than BEP and OvA, which are optimized for small datasets on all datasets. We also observe that SNN does better than the Image dataset. However, CDAN-dependent performs comfortably better than all the other methods except for SR at lower rejection rates on the Image dataset. In addition, we make another important observation that DAC fails to reject any of the samples on smaller datasets.

5.7 Empirical Observations on Large Datasets

We evaluated the effectiveness of CDAN for large datasets by plotting accuracy vs. rejection rates computed over five iterations. We observed that the results obtained on large datasets are comparable with the other baseline methods. However, we notice a few critical observations. Firstly, the CDAN network requires an additional auxiliary task. Otherwise, the network converges to meager rejection rates (0–2%) or very high rejection rates(99–100%). Secondly, an advantage over methods parameterized by the cost of rejection, such as BEP and OvA, is that they fail to work due to computational restraints even on marginally large dimensional datasets. We also observe our network performs comparably to the state-of-the-art techniques SelectiveNet, DAC, and SR on large datasets and performs moderately better across the three datasets when the rejection rate is lower (<10%). In addition, we also observed that DAC and CDAN could achieve zero rejection rates. Still, SelectiveNet cannot achieve full coverage, and some post-processing needs to be done to achieve full coverage in SNN. The DAC loss function is independent of any other loss function; CDAN and SNN use a convex combination of cross-entropy loss and independent losses, while SR only uses cross-entropy loss.

6 Conclusions and Future Research Directions

We presented an effective deep-learning model for classification with abstention. This model also learns both the decision surface and bandwidth rejection parameters simultaneously. Novel loss function *Bounded Multiclass Abstention loss* is proposed in this paper. We empirically established the network's performance by comparing it against the state-of-the-art methods on small and large datasets. The results motivate the use of CDAN in critical applications where the cost of misclassification is high.

There are several directions for future research. On the explainability of rejection decisions, we can look further into the features learned by the model and how they differ from features learned by non-abstention networks. We would also like to explore the network's performance when noise is introduced into the datasets.

References

1. Bartlett, P.L., Wegkamp, M.H.: Classification with a reject option using a hinge loss. JMLR **9**(Aug), 1823–1840 (2008)

2. Chow, C.: On optimum recognition error and reject tradeoff. IEEE Trans. Inf. Theory **16**(1), 41–46 (1970)
3. Cortes, C., DeSalvo, G., Mohri, M.: Boosting with abstention. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
4. Cortes, C., DeSalvo, G., Mohri, M.: Learning with rejection. In: Proceedings of 27th Conference on Algorithmic Learning Theory (ALT), Bari, Italy, vol. 9925, pp. 67–82 (2016)
5. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2**, 265–292 (2002)
6. da Rocha Neto, A.R., Sousa, R., de A. Barreto, G., Cardoso, J.S.: Diagnostic of pathology on the vertebral column with embedded reject option. In: Pattern Recognition and Image Analysis, pp. 588–595 (2011)
7. Dua, D., Graff, C.: UCI machine learning repository (2017)
8. El-Yaniv, R., et al.: On the foundations of noise-free selective classification. JMLR **11**(5) (2010)
9. Elson, J., Douceur, J., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: Proceedings of 14th ACM Conference on Computer and Communications Security (CCS) (2007)
10. Fumera, G., Pillai, I., Roli, F.: Classification with reject option in text categorisation systems. In: 2003 Proceedings of the 12th International Conference on Image Analysis and Processing, pp. 582–587 (2003)
11. Geifman, Y., El-Yaniv, R.: Selective classification for deep neural networks. In: NIPS, pp. 4878–4887 (2017)
12. Geifman, Y., El-Yaniv, R.: SelectiveNet: a deep neural network with an integrated reject option. In: ICML, pp. 2151–2159 (2019)
13. Grandvalet, Y., Rakotomamonjy, A., Keshet, J., Canu, S.: Support vector machines with a reject option. In: NIPS, pp. 537–544 (2009)
14. Hanczar, B., Dougherty, E.R.: Classification with reject option in gene expression data. Bioinform. (Oxford Engl.) **24**(17), 1889–1895 (2008)
15. Kalra, B., Shah, K., Manwani, N.: RISAN: robust instance specific deep abstention network. In: Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence, vol. 161, pp. 1525–1534 (2021)
16. Kamiran, F., Mansha, S., Karim, A., Zhang, X.: Exploiting reject option in classification for social discrimination control. Inf. Sci. **425**(C), 18–33 (2018)
17. Kompa, B., Snoek, J., Beam, A.L.: Second opinion needed: communicating uncertainty in medical machine learning. NPJ Digit. Med. **4**(1), 1–6 (2021)
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. LeCun, Y., Cortes, C., Burges, C.J.: MNIST handwritten digit database. ATT Labs (2010). https://yann.lecun.com/exdb/mnist
20. Li, Q., Vempaty, A., Varshney, L.R., Varshney, P.K.: Multi-object classification via crowdsourcing with a reject option. IEEE Trans. Signal Process. **65**(4), 1068–1081 (2017)
21. Madras, D., Pitassi, T., Zemel, R.S.: Predict responsibly: improving fairness and accuracy by learning to defer. In: Neural Information Processing Systems (2017)
22. Manwani, N., Desai, K., Sasidharan, S., Sundararajan, R.: Double ramp loss based reject option classifier. In: Cao, T., Lim, E.-P., Zhou, Z.-H., Ho, T.-B., Cheung, D., Motoda, H. (eds.) PAKDD 2015. LNCS (LNAI), vol. 9077, pp. 151–163. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18038-0_12

23. Mesquita, D.P.P., Rocha, L.S., Gomes, J.P.P., Rocha Neto, A.R.: Classification with reject option for software defect prediction. Appl. Soft Comput. **49**, 1085–1093 (2016)
24. Mohseni, S., Pitale, M., Singh, V., Wang, Z.: Practical solutions for machine learning safety in autonomous vehicles. In: SafeAI@AAAI (2020)
25. Mozannar, H., Sontag, D.: Consistent estimators for learning to defer to an expert. In: Proceedings of the 37th ICML (2020)
26. Pietraszek, T.: Optimizing abstaining classifiers using roc analysis. In: ICML, pp. 665–672 (2005)
27. Ramaswamy, H.G., Tewari, A., Agarwal, S., et al.: Consistent algorithms for multiclass classification with an abstain option. Electron. J. Stat. **12**(1), 530–554 (2018)
28. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. **5**, 101–141 (2004)
29. Rosowsky, Y.I., Smith, R.E.: Rejection based support vector machines for financial time series forecasting. In: IJCNN, pp. 1–7 (2013)
30. Shah, K., Manwani, N.: Sparse reject option classifier using successive linear programming. In: AAAI, vol. 33, pp. 4870–4877 (2019)
31. Shah, K., Manwani, N.: Online active learning of reject option classifiers. In: AAAI, pp. 5652–5659 (2020)
32. Sridhar, K., Busso, C.: Speech emotion recognition with a reject option. In: INTERSPEECH (2019)
33. Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., Mohd-Yusof, J.: Combating label noise in deep learning using abstention. In: ICML, vol. 97, pp. 6234–6243 (2019)

Learning and Regression on the Grassmannian

Anis Fradi(✉) and Chafik Samir

University of Clermont Auvergne, LIMOS CNRS (UMR 6158),
63000 Clermont-Ferrand, France
{anis.fradi,chafik.samir}@uca.fr

Abstract. In this paper, we introduce a new method for learning and regression from a finite set of noisy points on Grassmann manifolds. In contrast to previously existing methods, we propose a new Riemannian Monte Carlo method to sample from the posterior distribution of the tangent space of a Grassmann manifold. Specifically, we investigate and exploit the geometric structure of this manifold which can be used as a solid basis to extend the proposed method to other manifolds in a similar manner. We demonstrate our method for regression using different setups and datasets.

Keywords: Regression Model · Grassmann Manifolds · Riemannian Monte Carlo · Manifold-valued Data

1 Introduction

Manifold-valued data [2] are often encountered in various fields such as computer vision [10], natural language processing [12], and shape analysis [17]. Some examples include images, speech signals, and landmarks. Traditional learning methods may not be effective when dealing with manifold valued data. To address this challenge, several regression models taking into account the intrinsic geometry of the manifold have been developed [15,19]. Geodesic regression models as an extension of linear regression models are designed to handle data that resides on nonlinear manifolds. In linear regression, the relationship between the response variable and the predictors is assumed to be linear, and the least squares method is established to estimate the model parameters. In contrast, the geodesic regression deals with the concept of geodesics, which are the shortest paths on a manifold. By learning a geodesic regression model one can capture the underlying nonlinear relationship between the input and target variables. Geodesic regression models have been applied to a wide range of problems, including medical imaging, neuroscience, and computer vision.

The Grassmann manifold, denoted by $Gr(n, p)$, is a mathematical concept that represents the collection of all linear subspaces of a fixed dimension p in a given vector space $\mathbb{R}^n$ [3,20]. Geodesic regression on the Grassmannian is an

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 66–78, 2024.
https://doi.org/10.1007/978-981-99-7022-3_6

active area of research with several applications [5]. Firstly, [13] proposed a gradient method for geodesic data fitting on some symmetric Riemannian manifolds, which is still applicable for Grassmann manifolds. Secondly, [18] addressed the problem of estimating full curves/paths on quotient spaces of matrix nonlinear manifolds, using only a set of time-indexed points. Recently, [8] discussed extensions of linear and cubic spline regression on the Grassmannian within an optimal-control perspective. More recently, there has been interest in applying deep learning techniques to geodesic regression on the Grassmannian. For example, [9] used a neural network to estimate geodesic regression models on the Grassmannian.

In literature, there are various avenues for generating samples from complex posterior distributions with Markov chain Monte Carlo (MCMC) methods [7], ranging from simple Metropolis Hasting (MH) methods with symmetric proposal distributions to component-wise Gibbs samplers. The basic principle of MCMC is to construct a Markov chain trajectory whose stationary distribution is the desired probability distribution. The Markov chain is constructed in such a way that it satisfies the detailed balance condition, which ensures that the chain will converge to the desired probability distribution [6]. One of the main advantages of Monte Carlo sampling is that it can be used to sample from distributions that are not analytically tractable, such as posterior distributions in Bayesian statistics. Additionally, MCMC can handle non-convex and multi-modal distributions, which are difficult to sample using other basic techniques.

In this paper, we propose to learn a regression model on the Grassmannian based on a Riemannian Monte Carlo sampling that could overcome some limitations of deterministic optimization algorithms [14]. The main advantage is its ability to efficiently explore the manifold and avoid getting trapped in local minima. This is because Riemannian Monte Carlo algorithms can sample from the posterior distribution of the Grassmannian's tangent space, which provides a rich set of candidate directions [11]. They can also handle noise and uncertainty in data by incorporating prior knowledge and uncertainty into the optimization process. This can improve the robustness and accuracy of the underlying model [1]. Additionally, the proposed algorithm can be easily parallelized, which allows for efficient exploration of the manifold and speed up the learning step. Overall, the ability of the Riemannian Monte Carlo sampling to efficiently explore the manifold, handle noise and uncertainty, and parallelize well, make it a powerful tool for Riemannian optimizations.

Organization. The paper is organized as follows: Sect. 2 introduces some background about the geometry of the Grassmann manifold in connection with the Stiefel representation. In Sect. 3 we present our proposed method about the regression on Grassmannian as well as the corresponding algorithm. Section 4 then presents experimental results whereas Sect. 5 concludes the paper with a discussion and an outlook on future work.

2 The Geometric Structure

The set of $n \times p$ matrices with p-dimensional orthogonal columns in $\mathbb{R}^n$, is known as the real Stiefel manifold denoted by $St(n,p)$, which is a (compact) Riemannian manifold of dimension $np - \frac{1}{2}p(p+1)$ satisfying

$$St(n,p) = \{U \in \mathbb{R}^{n\times p} \mid U^T U = I_p\}. \tag{1}$$

We define the orthogonal group by

$$O(p) = \{R \in \mathbb{R}^{p\times p} \mid R^T R = RR^T = I_p\}. \tag{2}$$

We denote the action of $O(p)$ on $St(n,p)$ by the right multiplication as an equivalence class

$$[U] = \Big\{UR \mid R \in O(p)\Big\}. \tag{3}$$

We remark that, up to right rotations $R \in O(p)$, the mapping $U \mapsto [U]$ maps an element of the Stiefeld manifold $St(n,p)$ to an element of the quotient space $St(n,p)/O(p)$. This quotient space is in one-to-one correspondence with the set of p-dimensional linear subspaces of $\mathbb{R}^n$, namely the Grassmann manifold, satisfying

$$Gr(n,p) = \Big\{\mathcal{U} = [U] \mid U \in St(n,p)\Big\}, \tag{4}$$

according to $Gr(n,p) \cong St(n,p)/O(p)$. Note that the dimension of the Grassmann manifold is $n(n-p)$. Let $T_{\mathcal{U}}Gr(n,p)$ denote the tangent space of $Gr(n,p)$ locally at $\mathcal{U}$. The canonical Riemannian metric $g_{\mathcal{U}} : T_{\mathcal{U}}Gr(n,p) \times T_{\mathcal{U}}Gr(n,p) \to \mathbb{R}$ on $Gr(n,p)$ is defined by

$$g_{\mathcal{U}}(\varDelta_1, \varDelta_2) = \mathrm{tr}(\varDelta_1^T \varDelta_2). \tag{5}$$

Let U be an orthogonal class representative in $St(n,p)$ with columns spanning $\mathcal{U}$ i.e., $[U] = \mathcal{U}$. Then the projection of an arbitrary matrix $M \in \mathbb{R}^{n\times p}$ onto the tangent space at $\mathcal{U}$ is a tangent vector $\varDelta$ such that

$$\varDelta = (I_n - UU^T)M. \tag{6}$$

Therefore, the canonical Riemannian metric between two tangent vectors $\varDelta_1$ and $\varDelta_2$ becomes

$$g_{\mathcal{U}}(\varDelta_1, \varDelta_2) = \mathrm{tr}(M_1^T (I_n - UU^T) M_2). \tag{7}$$

Geodesic paths on the Grassmannian $Gr(n,p)$ denoted γ are locally the shortest curves between two points that are parametrized by the arc length. Moreover, exponential maps $Exp_{U_1} : T_{\mathcal{U}_1}Gr(n,p) \to Gr(n,p)$ maps a tangent vector $\varDelta \in T_{\mathcal{U}_1}Gr(n,p)$ to the end point of the geodesic γ. To summarize, exponentials and geodesic paths are related by $\gamma(t) = Exp_{U_1}(t\varDelta)$. Consider a curve $\gamma : [0,1] \to$

$Gr(n,p)$; $t \mapsto \gamma(t)$ such that $\gamma(0) = U_1$ and $\gamma(1) = U_2$. The geodesic equation for such curve on $Gr(n,p)$, given that $\dot{\gamma} = \frac{d}{dt}\gamma(t) = (I_n - UU^T)M$, is

$$\ddot{\gamma}(t) + \gamma(t)\big(\dot{\gamma}(t)^T\dot{\gamma}(t)\big) = 0. \tag{8}$$

The Grassmann geodesic path, as a solution of the geodesic equation [5], starting at $\gamma(0) = U_1$ with a direction $\varDelta \in T_{\mathcal{U}_1}Gr(n,p)$ is

$$\gamma(t) = U_1 V \cos(t\varSigma)V^T + Q\sin(t\varSigma)V^T, \tag{9}$$

with $\varDelta \overset{\text{SVD}}{:=} Q\varSigma V^T$, $Q \in St(n,p)$, $\varSigma \in diag(\mathbb{R}^{p\times p})$ and $V \in O(p)$. Moreover, the arc-length of the Grassmann geodesic path connecting two points $\mathcal{U}_1 = [U_1]$ and $\mathcal{U}_2 = [U_1] \in Gr(n,p)$ is related to the canonical angles $\varPhi = (\phi_1,\ldots,\phi_p)^T \in [0,\pi/2]$ between U_1 and U_2 according to $\text{dist}(U_1,U_2) = ||\varPhi||_2$. This is the geodesic distance which can be computed from the SVD decomposition $U_1^T U_2 = Q\cos(\varSigma)V^T$ (where $\varSigma$ is a diagonal matrix with principle angles ϕ_j) as

$$\text{dist}(U_1,U_2) = ||\cos^{-1}(diag(\varSigma))||, \tag{10}$$

where $||.||$ denotes the Frobenius norm induced by the trace inner product. This shows that, for any two points $\mathcal{U}_1$ and $\mathcal{U}_2$ on the Grassmann manifold $Gr(n,p)$ represented in the Stiefel level by U_1 and U_2, the geodesic distance is bounded by

$$\text{dist}(U_1,U_2) \leq \sqrt{p}\frac{\pi}{2}. \tag{11}$$

The inverse exponential map (log-map) on the Grassmannian $Log_{U_1} : Gr(n,p) \to T_{\mathcal{U}_1}Gr(n,p)$ is a diffeomorphism that maps a neighborhood of U_1 to $T_{\mathcal{U}_1}Gr(n,p)$. We write

$$Log_{U_1}(U_2) = \varDelta \quad if \quad Exp_{U_1}(\varDelta) = \gamma(1) = U_2. \tag{12}$$

Using the SVD decomposition $(U_2 - U_1U_1^TU_2)(U_1^TU_2)^{-1} = Q\varSigma V^T$ yields that

$$\varDelta = Q\arctan(\varSigma)V^T. \tag{13}$$

Let $\mathcal{U}_1,\mathcal{U}_2 \in Gr(n,p)$, and a direction matrix $\varDelta_1 \in T_{\mathcal{U}_1}Gr(n,p)$ such that $U_2 = \gamma(1) = Exp_{U_1}(\varDelta_1)$. The parallel transport consists of transporting an arbitrary tangent vector $\varDelta_2 \in T_{\mathcal{U}_1}Gr(n,p)$ to $T_{\mathcal{U}_2}Gr(n,p)$ along the geodesic connecting U_1 and U_2, satisfying

$$\varGamma_{U_1\to U_2}(\varDelta_2) = -U_2V\sin(\varSigma)Q^T\varDelta_2 + Q\cos(\varSigma)Q^T\varDelta_2 + (I_n - QQ^T)\varDelta_2, \tag{14}$$

based on the SVD decomposition $\varDelta_1 = Q\varSigma V^T$.

3 The Regression Model

It is important to mention that there are many other representations of the Grassmann manifold. For instance, it can be seen as a quotient space of the orthogonal group $O(n)$ from $Gr(n,p) \cong O(n)/\big(O(n-p) \times O(p)\big)$ or a quotient space of the "noncompact Stiefel manifold" (the set of all $n \times p$ matrices whose columns are linearly independent) denoted $\mathbb{R}_*^{n\times p}$ from $Gr(n,p) \cong \mathbb{R}_*^{n\times p}/GL_p$ where GL_p denotes the set of all $p \times p$ invertible matrices [1]. However, in this paper, we prefer its connection to the Stiefel manifold $Gr(n,p) \cong St(n,p)/O(p)$ for which the geodesic distance, parallel transport, as well as the Riemannian Exp-map and its inverse are relatively simple to compute (9 → 14).

3.1 Formulation

In this section, we introduce a geodesic regression model for the Grassmann manifold. Let $(t_i, y_i)_{i=1}^N$ be a finite set of measurements in the Grassmann manifold $Gr(n,p)$. For simplicity and without loss of generality, we assume that the input t_i is a time instance in $[0,1]$ and y_i refers to its corresponding output in $Gr(n,p)$. The regression captures the relationship between data points on the Grassmannian y_i and their associated independent variables t_i. We remind that the geodesic distance, induced by this metric, was denoted as dist(., .) in (10).

Definition 1. *A random matrix $X \in \mathbb{R}^{n\times p}$ is said to have a matrix-variate Gaussian distribution that we denote $X \sim MN(m|C_1, C_2)$ if and only if its probability density function is given by*

$$Pr(X|m, C_1, C_2) = (2\pi)^{-\frac{np}{2}} det(C_1)^{-\frac{n}{2}} det(C_2)^{-\frac{p}{2}} \times e^{-\frac{1}{2} tr\left(C_2^{-1}(X-m)^T C_1^{-1}(X-m)\right)}, \tag{15}$$

with a mean matrix $m \in \mathbb{R}^{n\times p}$ and positive semi-definite covariance matrix $C_1 \in \mathbb{R}^{n\times n}$, $C_2 \in \mathbb{R}^{p\times p}$ where det(.) is the matrix determinant.

We define the regression model as

$$y_i = Exp_{\gamma(0)}\big(Log_{\gamma(0)}(\gamma(t_i)) + \epsilon_i\big); \quad \epsilon_i \sim MN(0|\sigma I_n, \sigma I_p); \quad i = 1, \ldots, N \tag{16}$$

where ϵ_i is a matrix-variate Gaussian noise realization assumed to be added on the tangent space at $\gamma(0)$. Here, $\gamma(t)$ refers to the Grassmann geodesic path at initial point $\gamma(0)$ with an initial direction $\dot\gamma(0) \in T_{[\gamma(0)]}Gr(n,p)$. From (9) yields

$$\gamma(t) = Exp_{\gamma(0)}(t\dot\gamma(0)) = \gamma(0)V\cos(t\Sigma)V^T + Q\sin(t\Sigma)V^T, \tag{17}$$

for $\dot\gamma(0) \overset{\text{SVD}}{:=} Q\Sigma V^T$. The main objective is to estimate the geodesic path $\gamma : [0,1] \to Gr(n,p)$ entirely determined by initial conditions: the initial position $\gamma(0)$ and the initial direction $\dot\gamma(0)$. The injectivity radius at any $\mathcal{U} \in Gr(n,p)$ is defined as the distance from $\mathcal{U}$ to its cut locus, or the radius r for which

$Exp_U(.)$ is a diffeomorphism from the open ball $B_r(0) \subset T_{\mathcal{U}}Gr(n,p)$ into its image. From (11) we state that the injectivity radius is $r = \frac{\pi}{2}$ since there is always a subspace for which the principal angles between a first point U_1 and a second one on the cut locus U_2 are all equal to zero, except one, which is equal to $\frac{\pi}{2}$. Consequently, the noise realization ϵ_i should be controlled in such a way that $||Log_{\gamma(0)}(\gamma(t_i)) + \epsilon_i|| < \frac{\pi}{2}$ allowing the model in (16) to be well-defined.

3.2 Inferring on the Optimal Model

The likelihood is the probability of measurements $y_1, ..., y_N$ in $Gr(n,p)$ satisfying

$$Pr(y_1, \ldots, y_N | t_1, \ldots, t_N, \gamma(0), \dot{\gamma}(0)) \propto e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{N} \mathrm{dist}^2\big(y_i, \gamma(t_i)\big)}. \tag{18}$$

Maximizing the likelihood above can be reformulated as a minimization problem

$$\min_{\{\theta,v\}} E(\theta, v) := \frac{1}{2\sigma^2}\sum_{i=1}^{N} \mathrm{dist}^2\big(y_i, \gamma(t_i)\big), \tag{19}$$
$$s.t. \quad \{\theta, v\} = \{\gamma(0), \dot{\gamma}(0)\}; \quad \theta^T\theta = I_p \quad \text{and} \quad \theta \perp v.$$

The two initial conditions θ and v correspond to intercept and slope in linear regression. The goal is then to find a geodesic curve $t \mapsto \hat{\gamma}(t) = \gamma(t, \hat{\theta}, \hat{v})$ that minimizes the sum of the squared Riemannian distances between the data points y_i and their corresponding points on the geodesic curve $\gamma(t_i)$. On the one hand, by applying the Logarithm map at $\gamma(0)$ to (16) we have $Log_{\gamma(0)}(y_i) = Log_{\gamma(0)}(\gamma(t_i)) + \epsilon_i$ which implies that $\epsilon_i = Log_{\gamma(0)}(y_i) - Log_{\gamma(0)}(\gamma(t_i)) = Log_{\gamma(t_i)}(y_i)$. This means that the i-th residual measurement ϵ_i results to be the Log-map of y_i into the the tangent space at $\gamma(t_i)$. On the other hand, we state that $\mathrm{dist}\big((y_i, \gamma(t_i)\big) = ||Log_{\gamma(0)}(y_i) - Log_{\gamma(0)}(\gamma(t_i))||$ since the Log-map is a Riemannian isometry from the Grassmannian to its tangent space. Consequently, we get $\mathrm{dist}^2\big(y_i, \gamma(t_i)\big) = ||\epsilon_i||^2$. Therefore, the maximum likelihood estimate (MLE) of (θ, v) coincides with the least-squares solution in the Euclidean tangent space obtained by minimizing the loss function $E(\theta, v) = \frac{1}{2\sigma^2}\sum_{i=1}^{N} ||\epsilon_i||^2$.

Many methods have been proposed to deal with data that possess a dynamic behavior. For instance, [13] proposed a method based on a gradient descent technique on the tangent bundle of the Grassmannian. In addition, [8] extended the basic Riemannian optimization to time-warped variants and cubic splines. In contrast to these methods, our idea consists of adding a prior information on θ and v in a Bayesian framework with two prior matrix-variate Gaussian distributions

$$\theta \sim MN(0|C_{1,\theta}, C_{2,\theta}) \times \mathbf{1}_{\{\theta^T\theta = I_p\}}, \tag{20}$$
$$v \sim MN(0|C_{1,v}, C_{2,v}) \times \mathbf{1}_{\{\theta^T v = 0_p\}}, \tag{21}$$

where $\mathbf{1}_{\{\theta^T\theta = I_p\}}$ and $\mathbf{1}_{\{\theta^T v = 0_p\}}$ denotes the indicator function restricting θ to be an orthogonal matrix and v to be its corresponding tangent vector, respectively.

Algorithm 1. Grassmann Riemannian Monte Carlo (GRMC) sampling.

```
Require: $(t_i, y_i)_{i=1}^N$
  Initialize $\theta^0$ and $v^0$
  for $l = 0, 1, 2, \ldots, iter$ do
      Define $\theta = \theta^{(l)}$ and $v = v^{(l)}$
      Generate a velocity $v^* \sim q(.)$ from a proposal distribution $q$
      Project simulated velocity into the tangent space (for consistency)
          $v^* \leftarrow (I_n - \theta\theta^T)v^*$
      Map $v$ into the Grassmannian locally at $\theta$
          $\theta' \leftarrow Exp_\theta(v) = \theta V \cos(\Sigma)V^T + Q\sin(\Sigma)V^T$   with   $v \overset{\text{SVD}}{:=} Q\Sigma V$
      Transport $v^*$ along the geodesic connecting $\theta$ to $\theta'$
          $v' \leftarrow -\theta' V \sin(\Sigma)Q^T v^* + Q\cos(\Sigma)Q^T v^* + (I_n - QQ^T)v^*$
      Compute the acceptance probability
          $\alpha = \frac{Pr(\theta', v')}{Pr(\theta, v)} \frac{q(v)}{q(v')}$   where   $Pr(.,.)$is the posterior distribution in (22)
      Evaluate $r = \min(\alpha, 1)$
      Generate $u \sim U(0,1)$
      if $u \leq r$ then
          $\theta^{(l+1)} \leftarrow \theta'$ and $v^{(l+1)} \leftarrow v'$
      else
          $\theta^{(l+1)} \leftarrow \theta^{(l)}$ and $v^{(l+1)} \leftarrow v^{(l)}$
      end if
  end for
```

From Bayes' rule the posterior on (θ, v) satisfies

$$\begin{aligned} Pr(\theta, v | y_1, \ldots, y_N, t_1, \ldots, t_N) &\propto Pr(y_1, \ldots, y_N | t_1, \ldots, t_N, \theta, v) \\ &\quad \times Pr(\theta | 0, C_{1,\theta}, C_{2,\theta}) \times Pr(v | 0, C_{1,v}, C_{2,v}) \\ &\propto e^{-\frac{1}{2\sigma^2}\sum_{i=1}^N \text{dist}^2\left(y_i, \gamma(t_i)\right)} \times e^{-\frac{1}{2} tr\left(C_{2,\theta}^{-1}\theta^T C_{1,\theta}^{-1}\theta\right)} \\ &\quad \times \mathbf{1}_{\{\theta^T\theta = I_p\}} \times e^{-\frac{1}{2} tr\left(C_{2,v}^{-1} v^T C_{1,v}^{-1} v\right)} \times \mathbf{1}_{\{\theta^T v = 0_p\}}. \end{aligned} \tag{22}$$

Now, we are ready to maximize the posterior distribution subject to (θ, v) to get the maximum a posteriori (MAP) estimate. The Grassmann Riemannian Monte Carlo (GRMC) described in Algorithm 1 samples from distributions with Grassmannian structures. However, it can be computationally intensive and require careful tuning of the proposal distribution to ensure an efficient convergence to the target distribution.

3.3 A Specific Example of the Grassmannian

Affine invariance refers to the property of an object or shape that remains the same even after applying an affine transformation, such as rotation, scaling, and translation [4]. Landmark shapes, which are commonly used in computer vision and pattern analysis, often require affine invariance to ensure accurate and reliable analysis [16]. One way to achieve affine invariance for landmark shapes is by using SVD or QR decomposition. Both decompositions can be used to represent the landmark shapes as a linear combination of a few basis shapes, which can be maintained to reconstruct the original shape with affine invariance. If we have a set of planar shapes represented as 2D points in a matrix $L = [(x_1^1, x_1^2); (x_2^1, x_2^2); \ldots ; (x_n^1, x_n^2)]$, where each row represents a landmark point and each column represents one direction we can use SVD or QR decomposition to factorize L into a product of three or two matrices, such that $L = Q\Sigma V^T$ or $L = QR$, respectively. In both cases, the matrix Q contains the orthonormal basis vectors of L, while the matrix Σ or R contains scaling information about each basis vector. This establishes a mapping from the shape matrix to a point on the Grassmannian (with Q as a Stiefel representation). Such representation has been used for a wide range of applications, including image classification, image registration, and object tracking.

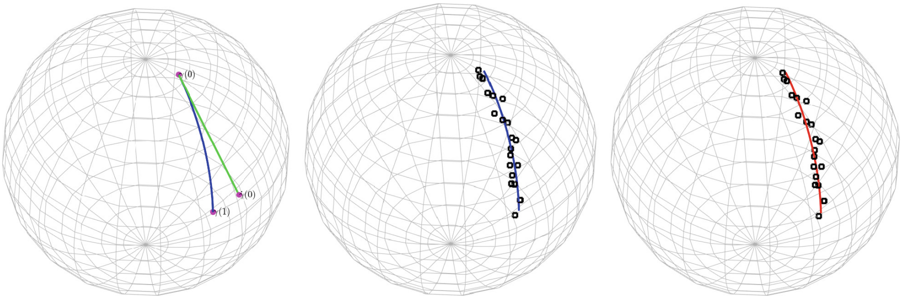


Fig. 1. An illustration on $Gr(3,1)$. From left to right: Initial conditions $\gamma(0)$, $\gamma(1) = Exp_{\gamma(0)}(\dot{\gamma}(0))$ in pink with true geodesic path in blue and tangent space in green, noisy observations in black, and predicted path in red. (Color figure online)

4 Experimental Results

In order to validate the efficacy of our approach, we conducted series of experiments on various dataset and present results in this section. We employ the GRMC sampling to generate samples from the posterior distribution of the Bayesian model with $iter = 10^5$ iterations. At each iteration, we propose a new candidate using a proposal distribution that was designed to move the chain through the parameter space in a way that preserved detailed balance. Specifically, we use a Gaussian random-walk proposal distribution $q(.)$ on the tangent space of the Grassmannian with a fixed variance. There are several frequently employed measures for assessing the effectiveness of our model, which include:

- Mean squared geodesic distance (MSGD): the average squared geodesic distance between the predicted values and the observed values on the Grassmannian, providing a quantitative measure of the model' accuracy.

- R-squared: measures the variation in the dependent variable that can be explained by the independent variables in the model.
- Data-to-noise-ratio (DNR): quantifies the amount of useful information in data relative to the amount of noise.

To validate the effectiveness of our proposed methodology, we performed a comprehensive comparison against two existing deterministic techniques focusing on regression problems on the Grassmannian: i) the gradient method (GM) for geodesic data fitting [13] and ii) the standard Grassmannian geodesic regression (Std-GGR) [8].

Synthetic Data. We consider two examples of simulated data on $Gr(3, 1)$ and $Gr(3, 2)$ represented in the Stiefel level $St(3, 1)$ and $St(3, 2)$, respectively. Note that $St(3, 1)$ results to be the unit two-dimensional sphere denoted S^2 while $St(3, 2)$ can be also considered as the unit sphere, modulo rigid transformations, usually called the Kendall space. The time instants are uniformly spaced in $[0, 1]$ using $N = 20$ points t_i; $i = 1, \ldots, N$. The two initial conditions $\gamma(0)$ and $\dot{\gamma}(0)$ are chosen randomly to cover a large range of configurations. The full geodesic path $\gamma(t)$ depending on those conditions is then obtained from (17). We then simulate data from the following model

$$y_i = Exp_{\gamma(0)}\big(Log_{\gamma(0)}(\gamma(t_i)) + \epsilon_i\big); \quad \epsilon_i \sim MN(0|\sqrt{0.1}I_3, \sqrt{0.1}I_p); \quad p = 1, 2$$

An example of initial conditions when $p = 1$ is given in Fig. 1 (left). The true geodesic path is then corrupted with a Gaussian noise in Fig. 1 (middle). The result of Fig. 1 (right) shows that the learned path closely approximated the underlying path mainly in the presence of noise. Let $\hat{\gamma}(0)$ and $\hat{\dot{\gamma}}(0)$ denote the MAP estimates of $\gamma(0)$ and $\dot{\gamma}(0)$ obtained from Algorithm 1. Table 1 indicates that the proposed method achieves superior performance compared to other state-of-the-art methods in these experiments.

The second experiment concerns the regression problem where observations are elements of the Grassmannian with $(n, p) = (3, 2)$. Figure 2 (top) shows some true observations (in blue), along with a set of noisy observations (in interrupted black) sampled from the path. In Fig. 2 (bottom) we predict the same data (in red) using the sampling algorithm, which aims to maximize the posterior distribution in the Grassmann metric space. The learned path closely follows the observations and captures the underlying structure of the true path, despite the presence of noise. Table 2 demonstrates the effectiveness of our approach for modeling and analyzing complex data on Grassmann manifolds.

Real Data. We conducted several experiments using a real data to investigate the degeneration of the corpus callosum, collected from $N = 32$ subjects, represented as planar shapes and scaled through the QR decomposition. Each shape is comprised of $n = 64$ of 2D landmarks $(p = 2)$ and is accompanied by the age of the subject, which ranges from 19 to 90 years old. Overall, main results in Fig. 3 suggest that the corpus callosum over age has a significant deformation on the brain, and our regression model provides a reliable way to make prediction.

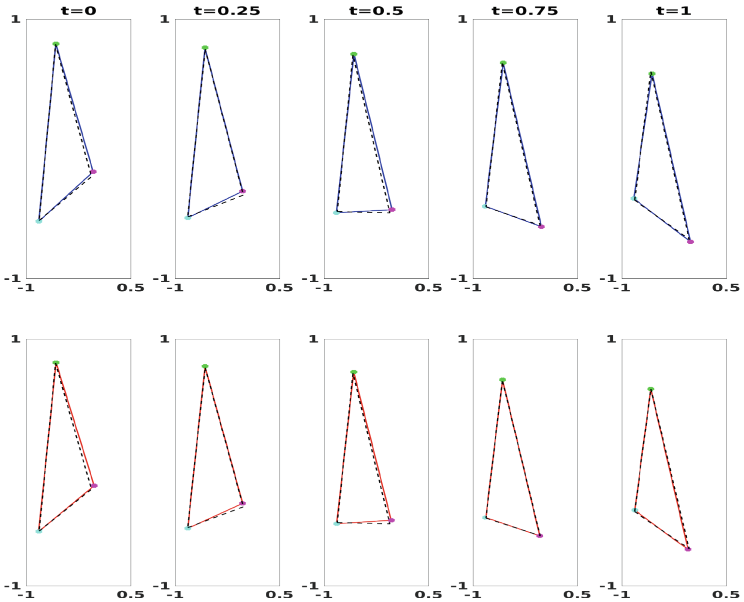


Fig. 2. An illustration on $Gr(3,2)$. (Top) True observations in blue with corner points taking the rows of $\gamma(t_i)$ and noisy observations in black and (Bottom) predicted ones with corner points taking the rows of $\hat{\gamma}(t_i)$ in red. (Color figure online)

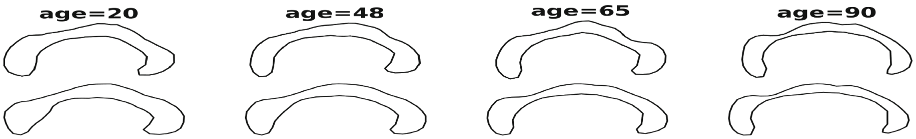


Fig. 3. (top) Four examples of observed corpus callosum and (bottom) their associated predictions among a list of planar shapes from age 19 to 90 years old.

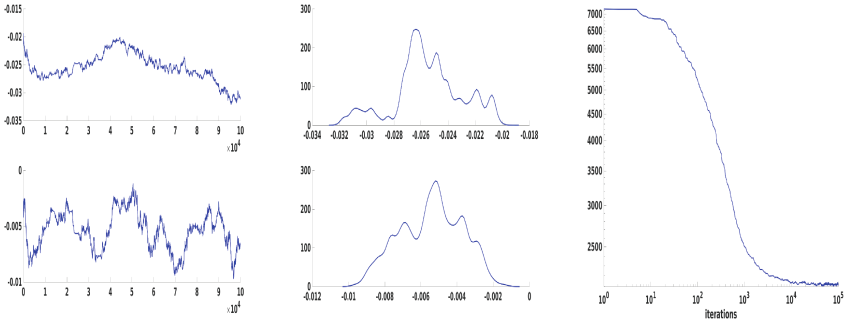


Fig. 4. (left) Trajectory of GRMC sampling: The (1, 1) element of the position $\gamma(0)$ (top) and the (1, 1) element of the velocity $\dot{\gamma}(0)$ (bottom), (middle) their kernel density estimations, and (right) the total energy loss function illustrated in a base-10 logarithmic scale.

Regarding Table 3 our proposed method achieves the best performances on two among three criteria: R-squared and DNR, while it still competitive in terms of MSGD.

In Fig. 4 (left) we illustrate the Markov chain trajectory over the iterations particularly for the (1, 1) element of both $\gamma(0)$ and $\dot{\gamma}(0)$. We also apply the "burn-in" allowing the sampler to reach its stationary distribution as well as the "thinning" allowing to reduce the autocorrelation between samples and saves computational resources. The trace plot shows how the GRMC sampler is able to explore the stationary posterior distribution that we want to approximate based on the kernel density estimation (KDE) method in Fig. 4 (middle). Finally, Fig. 4 (right) shows how the loss function steadily decreases with each iteration, indicating that our method is effectively learning the parameters. This is strong evidence that our method is capable of achieving high performance.

Table 1. Results on Gr(3, 1).

Method	$\vert\gamma(0) - \hat{\gamma}(0)\vert$	$\vert\dot{\gamma}(0) - \hat{\dot{\gamma}}(0)\vert$	MSGD	R-squared	DNR
GM	0.018	0.06	0.002	0.979	49.29
Std-GGR	0.033	0.042	0.0016	0.984	64.69
GRMC	**0.017**	**0.026**	**0.0015**	**0.985**	**64.91**

Table 2. Results on Gr(3, 2).

Method	$\vert\gamma(0) - \hat{\gamma}(0)\vert$	$\vert\dot{\gamma}(0) - \hat{\dot{\gamma}}(0)\vert$	MSGD	R-squared	DNR
GM	N/A	N/A	N/A	N/A	N/A
Std-GGR	0.0416	0.0687	0.0014	0.959	24.98
GRMC	**0.0172**	**0.0638**	**0.0013**	**0.962**	**26.54**

Table 3. Results on Corpus callosum.

Method	MSGD	R-squared	DNR
GM	0.0156	0.2469	1.3278
Std-GGR	**0.0144**	0.3470	1.5314
GRMC	0.0147	**0.3560**	**1.5528**

5 Conclusion

We have proposed a new method to learn the best regression model on Grassmann manifolds. In particular, we have designed a new GRMC sampling algorithm that is simple and easy to implement. Moreover, this framework can be extensible to other Riemannian manifolds by adjusting the appropriate geometrical tools. Regarding the applicability, we have used several setups and datasets. From experiments, we can conclude that the proposed method could solve many problems with competitive accuracy when compared to some connected works.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, USA (2007)
2. Amari, S.I., Barndorff-Nielsen, O.E., Kass, R.E., Lauritzen, S.L., Rao, C.R.: Differential geometry in statistical inference. In: Lecture Notes-Monograph Series, pp. 163–216. Institute of Mathematical Statistics, Hayward (1987)
3. Batzies, E., Hüper, K., Machado, L., Leite, F.S.: Geometric mean and geodesic regression on Grassmannians. Linear Algebra Appl. **466**, 83–101 (2015)
4. Begelfor, E., Werman, M.: Affine invariance revisited. In: Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2087–2094. IEEE, New York (2006)
5. Bendokat, T., Zimmermann, R., Absil, P.A.: A Grassmann manifold handbook: basic geometry and computational aspects (2020)
6. Cowles, M.K., Carlin, B.P.: Markov Chain Monte Carlo convergence diagnostics: a comparative review. J. Am. Stat. Assoc. **91**, 883–904 (1996)
7. Golightly, A., Wilkinson, D.: Bayesian parameter inference for stochastic biochemical network models using particle Markov Chain Monte Carlo. Interface Focus **1**, 807–820 (2011)
8. Hong, Y., Kwitt, R., Singh, N., Davis, B., Vasconcelos, N., Niethammer, M.: Geodesic regression on the Grassmannian. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 632–646. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_41
9. Huang, Z., Wu, J., Van Gool, L.: Building deep networks on Grassmann manifolds. In: AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 3279–3286 (2018)
10. Klette, R.: Concise Computer Vision - An Introduction into Theory and Algorithms. Undergraduate Topics in Computer Science. Springer, Heidelberg (2014). https://doi.org/10.1007/978-1-4471-6320-6
11. Livingstone, S., Girolami, M.: Information-geometric Markov chain Monte Carlo methods using diffusions. Entropy **16**, 3074–3102 (2014)
12. Radev, D.R., Muthukrishnan, P., Qazvinian, V., Abu-Jbara, A.: The ACL anthology network corpus. Lang. Resour. Eval. **47**, 919–944 (2013)
13. Rentmeesters, Q.: A gradient method for geodesic data fitting on some symmetric Riemannian manifolds. In: IEEE Conference on Decision and Control and European Control Conference, pp. 7141–7146 (2011)
14. Samir, C., Absil, P.A., Srivastava, A., Klassen, E.: A gradient-descent method for curve fitting on Riemannian manifolds. Found. Comput. Math. **12**, 49–73 (2012)

15. Sato, H.: Riemannian Optimization and Its Applications. Springer Briefs in Electrical and Computer Engineering. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-62391-3
16. Slice, D.: Landmark coordinates aligned by procrustes analysis do not lie in Kendall's shape space. Syst. Biol. **50**, 141–149 (2001)
17. Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.H.: Shape analysis of elastic curves in Euclidean spaces. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 1415–1428 (2011)
18. Su, J., Dryden, I., Klassen, E., Le, H., Srivastava, A.: Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. Image Vision Comput. - IVC **30**, 428–442 (2012)
19. Tripuraneni, N., Flammarion, N., Bach, F.R., Jordan, M.I.: Averaging stochastic gradient descent on Riemannian manifolds. In: Proceedings of Machine Learning Research, vol. 75, pp. 650–687. PMLR (2018)
20. Wong, Y.C.: Differential geometry of Grassmann manifolds. Proc. Natl. Acad. Sci. **57**, 589–594 (1967)

Partial Multi-label Learning with a Few Accurately Labeled Data

Haruhi Mizuguchi[1](✉), Keigo Kimura[1], Mineichi Kudo[1], and Lu Sun[2]

[1] Division of Computer Science and Information Technology, Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan
{mizuguchi-haruhi, kimura5, mine}@ist.hokudai.ac.jp

[2] School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
sunlu1@shanghaitech.edu.cn

Abstract. Partial Multi-label Learning is a multi-label classification problem where only candidate labels are given for training data. These candidate labels consist of relevant labels and false-positive labels. In this paper, we consider the PML when a few accurately labeled data are available. In practice, it is difficult to remove false-positive labels fully due to a large cost, but it is possible to do that in a few instances with a smaller cost. Conventional PML methods do not assume those accurately labeled data so it is hard to utilize data effectively. We propose a new algorithm called PML-VD to utilize those accurately labeled data. PML-VD first disambiguates the noisy-labeled data with both accurately labeled data and noisy labeled data and then learns a classifier. This two-stage approach enables the effective utilization of accurately labeled data without overfitting. Experiments on nine PML datasets shows the effectiveness of explicit utilization of accurately labeled data. In best cases, PML-VD improves 7% classification accuracy in terms of ranking loss.

Keywords: Partial Multi-label Learning · Multi-label Learning · Machine Learning · Ranking Loss

1 Introduction

Multi-label learning is a classification problem where multiple labels are simultaneously associated with a single instance. Multi-label learning aims to learn a function to predict all labels associated with a new instance. Since all combinations of labels should be taken into consideration, multi-label learning is more challenging than traditional single-label learning [12]. Multi-label learning has been widely used in text classification [7], bioinformatics [2], and image annotation [1].

In conventional multi-label learning, it is assumed that all training instances are accurately labeled, but in practice, the training data may be compromised with false-positive labels and only a candidate set of labels are available. This

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 79–90, 2024.
https://doi.org/10.1007/978-981-99-7022-3_7

specific challenge, training instances associated with not correct labels but candidate labels, is referred to as Partial Multi-label Learning (PML) [8]. This PML is commonly encountered in crowd-sourcing scenarios where accurate labels may not be guaranteed to obtain.

In general, those false-positive labels give negative impacts on classification performances. Thus, conventional PML methods attempt to mitigate this impact [5,8]. If we could remove false-positive labels, this impact could potentially be further reduced. In practice, it is possible to remove false-positive labels in a few instances does not require a large cost. Nonetheless, conventional PML methods do not assume those accurately labeled data, thus, PML methods cannot utilize those data effectively.

The only previous work tackling PML with a few accurately labeled data is a meta-learning-based method called PML-MD [10]. PML-MD minimizes a weighted ranking loss with the label "confidence". This confidence represents the likelihood of a label being the correct label. It serves to mitigate the impact of false-positive labels by weighting the loss function. The key concept of PML-MD is the utilization of a few accurately labeled data for confidence estimation. PML-MD learns both confidence and a classifier in an iterative manner, the confidence estimation using a small amount of accurately labeled data can incrementally causes over-fitting and harm the classifier's performance.

In this paper, we propose a two-stage method to mitigate this over-fitting issue in PML-MD. The proposed method first learns the confidence based on the smoothness assumption, a principle frequently employed in semi-supervised learning. Then, it learns a multi-label classifier in accordance with the learned confidence. The proposed method considers a few accurately labeled data called "validation set" and noisy labeled data separately and explicitly in the estimation of the confidence. Therefore, it effectively utilizes the validation set. Figure 1 shows the difference between PML-VD and PML-MD. Empirical studies have demonstrated that the proposed method outperforms the state-of-the-art methods across various datasets.

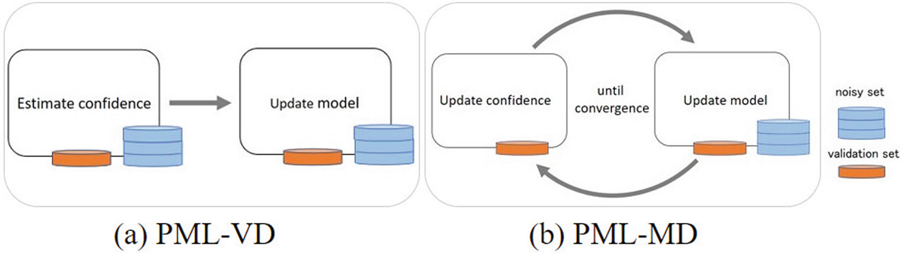


Fig. 1. An overview of PML-VD and PML-MD. (a) PML-VD first estimate confidence and second update the model. (b) PML-MD updates confidence and model iteratively.

2 Related Work

Existing PML methods can be broadly divided into two categories: end-to-end methods and two-stage methods. In the end-to-end methods, true labels are considered latent variables and are optimized iteratively with a classifier. PML-lc [8] and PML-fp [8] learn a classifier by minimizing the pairwise ranking loss weighted by confidence. PML-NI [9] assumes that false-positive labels are generated due to specific features and learns a multi-label classifier and a false-positive label identifier simultaneously to reduce the impact of noise labels. On the other hand, the two-stage methods divide the estimation of true labels and the learning of the classifier into two stages, as the end-to-end method is susceptible to the influence of false-positive labels. PARTICLE [11] estimates the confidence by a label propagation-like approach [13], and then learns a multi-label classifier based on the estimated true labels from the estimated confidence.

Those conventional PML methods have the uncertain assumption on labels for ambiguity resolution because they assume only candidate labels are given. Thus, those cannot utilize accurately labeled data even if available. PML-MD [10] is the only one previous research tackling the PML problem with a few accurately labeled data called a validation set. The aim of PML-MD is to use the validation set to estimate the confidence for noisy-labeled data through a meta-learning framework. PML-MD estimates the confidence and learns a classifier iteratively with a weighted pair-wise label ranking loss. In each iteration, it adjusts the confidence of noisy-labeled data with the validation set [10]. However, this method has a drawback, that is, it is prone to over-fitting a validation set through its iterative learning, especially when the size of the validation set is small. In this paper, we propose a two-stage method to mitigate the over-fitting issue in PML-MD due to its iterative learning.

3 Proposed Method

3.1 Problem Setting

In this paper, we consider a d-dimensional space $\mathcal{X} \subseteq \mathbb{R}^d$ for a feature vector $\boldsymbol{x} \in \mathcal{X}$ and a label space of q labels $\mathcal{Y} = \{0,1\}^q$ for a corresponding candidate label vector $\tilde{\boldsymbol{y}} \in \mathcal{Y}$. In PML, n training data points is given as $D = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^n$. Note that $\tilde{\boldsymbol{y}}$ is superfluously labeled, that is, $\tilde{\boldsymbol{y}}_i$ includes false-positive labels in addition to true labels. Thus, the relationship between a candidate label vector $\tilde{\boldsymbol{y}}$ and a true label vector $\boldsymbol{y}$ can be represented as $(\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\epsilon})$ where $\boldsymbol{\epsilon} \in \mathcal{Y}$ is a false-positive label vector.

We consider additional accurately labeled data. Thus, we introduce a set of m accurately labeled data points, distinct from D. This smaller set, referred to as the validation set, is denoted by $D_{val} = \{(\boldsymbol{x}_i^v, \boldsymbol{y}_i^v)\}_{i=1}^m$, with $(m \ll n)$. For clarity, hereafter, we refer to the training data D that excludes the validation set as the "noisy set." The goal of this study is learning a function $\boldsymbol{f}(\boldsymbol{x}, \theta) = \{f_1(\boldsymbol{x}, \theta), f_2(\boldsymbol{x}, \theta), \ldots, f_q(\boldsymbol{x}, \theta)\}$ with a parameter set θ to predict all labels from noisy set D and validation set D_{val}.

In this paper, we propose a two-stage method called PML-VD (Patial Mutlilabel Learning with Validation Data). PML-VD learns a classifier as follows:

1. Learning confidence which represents the likelihood of a label being the true label from both noisy set D and validation set D_{val}
2. Learning a classifier $\boldsymbol{f}(\boldsymbol{x}, \theta)$ with the learned confidence

This is the same approach as other two-stage PML methods, however, the proposed method explicitly utilizes validation set D_{val} on the confidence estimation.

3.2 Confidence Estimation with the Noisy Set and the Validation Set

PML-VD first estimates the confidence from the validation set and the noisy set. We define a confidence vector $\boldsymbol{p}_i$ where p_{ij} is the likelihood of j-th label being a true label of i-th instance. Estimating this confidence only from the noisy label set may harm the classification accuracy due to its false-positive labels. On the other hand, estimating only from the validation set also may harm the classification accuracy due to over-fitting. Therefore, PML-VD uses both the validation set and the noisy set. This is the key concept of the PML-VD.

In the confidence estimation, PML-VD takes into account the “smoothness assumption” which is commonly employed in semi-supervised learning, that is, the assumption that data points that are close together in feature space have the same labels. We consider applying this relationship in feature space to confidence. PML-VD learns this confidence from the validation set and noisy set separately.

First, we approximate an instance of D with instances of D_{val} linearly in feature space:

$$\boldsymbol{x}_i \simeq \sum_{j=1}^{m} w_{ij}^{v} \boldsymbol{x}_j^{v}, \quad \text{where } w_{ij}^{v} \geq 0. \tag{1}$$

Here, $w_{ij}^{v} \geq 0$ is introduced to ensure the non-negativity of confidence values obtained later. The weight w_{ij}^{v} is obtained by minimizing the following objective function:

$$\min_{w_{ij}^{v}} \ \|\boldsymbol{x}_i - \sum_{j=1}^{m} w_{ij}^{v} \boldsymbol{x}_j^{v}\|_2^2, \quad s.t. \ w_{ij}^{v} \geq 0. \tag{2}$$

This problem can be solved by the non-negative least square method [4]. Next, we estimate the confidence value with the learned weight w_{ij}^{v}. We use w_{ij}^{v} as the weight to calculate the confidence vector with true labels:

$$\hat{\boldsymbol{p}}_i{}^{v} = \sum_{j=1}^{m} w_{ij}^{v} \boldsymbol{y}_j^{v}, \quad (1 \leq i \leq n), \tag{3}$$

where $\hat{\boldsymbol{p}}_i^{v}$ is a confidence vector estimated from the validation set. Note that we estimate the confidence vector for each instance in the noisy set. The estimation

Algorithm 1. Partial Multi-label Learning with Validation Data

```
1: Input: A noisy set $D$:$\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$, a validation set $D_{val}$:$\{(\boldsymbol{x}_i^v, \tilde{\boldsymbol{y}}_i^v)\}_{i=1}^{m}$, An unseen instance $\boldsymbol{x}$, parameters $\alpha$, $k$ and maximum number of iterations $T$
2: Output: $f$
3: Estimate $\boldsymbol{w}^v$ for each $\boldsymbol{x} \in D$ linealy with all $\boldsymbol{x}^v \in D_{val}$ by solving (1).
4: Estimate $\boldsymbol{w}^n$ for each $\boldsymbol{x} \in D$ linealy with knn of $\boldsymbol{x}$ $\boldsymbol{x}_i \in D,\ i \in \kappa$ by solving (4).
5: Calculate the confidence $\boldsymbol{p}_i$ for each $\boldsymbol{x} \in D$ according to (3), (5), (6) and (7)
6: Initialize the parameter $\theta^{(0)}$
7: for $t = 1 : T$ do
8:     Sample a minibatch $D_b = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{b} \subseteq D + D_{val}$
9:     Update $\theta^{(t)}$ by minimize (10) with $D_b$
10: end for
```

may be inaccurate when the size of the validation set is small. Thus, PML-VD also estimates the confidence from the noisy set. This is done in the same manner as the validation set:

$$\boldsymbol{x}_i \simeq \sum_{j \in \kappa_i} w_{ij}^n \boldsymbol{x}_j, \quad \text{where } w_{ij}^n \geq 0. \tag{4}$$

Here, κ_i is the k-nearest neighbor index set of instance $\boldsymbol{x}_i$. We only introduce this k-nearest neighbor for the noisy set because of the smoothness assumption. Next, we estimate the confidence value with the learned weight w_{ij}^n:

$$\hat{\boldsymbol{p}_i}^n = \sum_{j \in \kappa_i} w_{ij}^n \tilde{\boldsymbol{y}}_j^n, \quad (1 \leq i \leq n), \tag{5}$$

where $\hat{\boldsymbol{p}_i}^n$ is a confidence vector estimated from the noisy set. At last, those two separately estimated confidence is combined with the weight $\alpha (0 \leq \alpha \leq 1)$:

$$\hat{\boldsymbol{p}}_i = (1 - \alpha)\hat{\boldsymbol{p}_i}^v + \alpha \hat{\boldsymbol{p}_i}^n. \tag{6}$$

Since PML does not assume false-negative labels, not-assigned labels in noisy sets are always negative, thus, confidence values can be fixed as:

$$\hat{p}_{ij} = \begin{cases} \hat{p}_{ij} & (\tilde{y}_{ij} = 1) \\ 0 & (\tilde{y}_{ij} = 0) \end{cases}. \tag{7}$$

This estimated confidence mitigates the influence of false-positive labels by weighting the loss function of a classifier, thereby facilitating the classifier's learning process.

3.3 Learning a Classifier

We use the weighted ranking loss proposed in [10] as an objective function of a classifier. This uses confidence (7) to mitigate the impact of false-positive labels. For the output of the classifier for each class of instance $\boldsymbol{x}_i$, $\boldsymbol{f}(\boldsymbol{x}_i,\theta) = \{f_1(\boldsymbol{x}_i,\theta), f_2(\boldsymbol{x}_i,\theta), \ldots, f_q(\boldsymbol{x}_i,\theta)\}$, the ranking loss is defined as follows:

$$l(f(\boldsymbol{x}_i,\theta),\boldsymbol{y}_i) = \sum_{j:y_{ij}=1}\sum_{k:y_{ik}=0} I\left[f_j\left(\boldsymbol{x}_i,\theta\right) < f_k\left(\boldsymbol{x}_i,\theta\right)\right], \tag{8}$$

where $I(\cdot)$ is the indicator function. This loss function represents the number of times irrelevant labels scored higher than relevant labels. Optimizing loss function (8) is NP-hard due to non-convexity and discreteness. Therefore, a convex surrogate loss is introduced for the optimization. A common surrogate loss for ranking loss is the following $Hingeloss$:

$$\mathcal{L}(f(\boldsymbol{x}_i,\theta),\boldsymbol{y}_i) = \sum_{j:y_{ij}=1}\sum_{k:y_{ik}=0} \ell\left(f_j\left(\boldsymbol{x}_i,\theta\right) - f_k\left(\boldsymbol{x}_i,\theta\right)\right), \tag{9}$$

where, $\ell(z) = \max(0, 1-z)$.

The classifier should score relevant labels higher than false-positive labels. However, since (9) treats all candidate labels as relevant labels, this could prevent the relevant labels from being scored higher than the false-positive labels. To mitigate this problem, the loss function is weighted by the difference in confidence between each label pair:

$$\mathcal{L}(D,\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_{ij}=1}\sum_{k:y_{ik}=0} \max\left(0, \hat{p}_{ij} - \hat{p}_{ik}\right)\ell\left(f_j\left(\boldsymbol{x}_i,\theta\right) - f_k\left(\boldsymbol{x}_i,\theta\right)\right), \tag{10}$$

$\max(0,\cdot)$ is introduced to prevent the loss from becoming negative. This makes the estimation of the confidence order more important for those with larger differences. Note that we used the classifier as employed in [10], however, PML-MD is classifier-agnostic, and thus, any classifier with the confidence such as VLS and MAP in PARTICLE [11] can be employed.

The computational complexity of the proposed PML-VD is $O(n^2 d)$ where n is the number of instances in the noisy set. This is because the k-nearest neighbor requires the calculation of similarities on the noisy set. On the other hand, the existing method PML-MD requires only $O(mnd)$ for the confidence estimation where m is the number of the validation set and $m \ll n$. Therefore, theoretically, PML-VD requires larger computational time than PML-MD. However, PML-MD needs to estimate confidence for each iteration, and thus, practically PML-VD often requires less computational time than PML-MD.

In summary, we proposed the PML method using a few accurately labeled data. Our approach divides the learning process into two stages: confidence estimation and classifier learning. In the first stage, we estimate the confidence by reflecting on the relationships in the feature space to the label space. During this stage, to supplement the estimations from the validation set, we also use the noisy set. In the second stage, we train a classifier by minimizing the ranking loss, weighted by the estimated confidence.

The pseudo-code of PML-VD is described in Algorithm 1.

Table 1. The summary of datasets. CLs means the average number of candidate labels. GLs means the average number of ground-truth labels.

Datasets	Domain	#Ins	#Fea	#Class	CLs	GLs
music_ emotion	image	6833	98	11	5.29	2.42
music_ style	image	6839	98	10	6.04	1.44
mirflickr	image	10433	100	7	3.35	1.77
emotions	image	593	72	6	–	1.87
enron	text	1702	1001	53	–	3.38
CAL500	music	502	68	174	–	26.04
scene	image	2407	294	6	–	1.07
genbase	biology	662	1186	27	–	1.25
yeast	biology	2417	103	14	–	4.24

4 Experiments

4.1 Experiments Setting

The summary of the dataset used in this paper is shown in Table 1. PML datasets are *music_ emotion*, *music_ style* and *mirflickr* [3]. These datasets come from an image search task [3], where candidate labels were assigned by web users and related labels were selected by the authors of [11]. For synthetic PML datasets, We employed the same setting in [10] to compare the proposed **PML-VD** with existing method **PML-MD** [10]. We converted six multi-label data into partial multi-label data. We constructed two variants ("high", "low") of the partial multi-label dataset for each dataset by flipping irrelevant labels. Those two are different in how noise labels are included:

1. High: each irrelevant class can be flipped by a randomly sampled probability. The probability is sampled from (0.5,0.6,0.7,0.8)
2. Low: each irrelevant class can be flipped by a randomly sampled probability. The probability is sampled from (0.2,0.3,0.4,0.5)

To verify the impact of the size of the validation set on the performance, we conducted experiments with quantities corresponding to {1, 3, 5, 7, 9}% of the total training data. We used *Ranking Loss* [10,11] as a measurement. This is because the classifier (10) returns a ranking of labels for each instance thus a binarization is required to obtain the binary classification result for each label. This is out of the scope of this paper.

To demonstrate the effectiveness of the proposed **PML-VD**, we compared with four PML methods, **PML-MD** [10], **PML-NI** [5], **PAR-VLS** [11] and **PAR-MAP** [11]. **PML-MD** is the only method to take the validation set into account explicitly. For the other method does not consider the validation set, the validation set and the noisy set were merged and given as training data. We also compared **Baseline** as multi-label learning methods. **Baseline** is the classifier explained in Sect. 3.3 regarding all candidate labels as true labels.

PML-VD has two hyperparameters k and α. We tuned them by 5-fold cross-validation on the training data. On the other hand, the hyperparameters for each comparative method were picked from their original paper. For **PML-NI**, we tuned its hyperparameters with 5-fold cross-validation as well.

4.2 Comparison Results

Table 2. Ranking loss on 1% validation set. The bold indicates the best method.

data	noise	PML-VD	PML-MD	Baseline	PML-NI	PAR-MAP	PAR-VLS
music_ emotion		**0.239**	0.242	0.247	0.242	0.362	0.265
music_ style		**0.137**	0.166	0.139	**0.137**	0.239	0.165
mirflickr		**0.066**	0.075	0.077	0.103	0.227	0.117
emotions	high	**0.270**	0.355	0.300	0.283	0.479	0.311
	low	**0.202**	0.292	0.224	0.220	0.469	0.216
enron	high	**0.163**	0.179	0.299	0.290	0.340	0.334
	low	0.163	**0.158**	0.267	0.210	0.316	0.260
CAL500	high	**0.282**	0.301	0.360	0.330	0.353	0.358
	low	**0.242**	0.263	0.288	0.254	0.271	0.346
scene	high	**0.144**	0.293	0.269	0.228	0.509	0.211
	low	**0.112**	0.236	0.172	0.126	0.471	0.119
genbase	high	**0.020**	0.238	0.044	0.022	0.462	0.104
	low	0.007	0.210	0.026	**0.006**	0.325	0.039
yeast	high	**0.218**	0.260	0.318	0.282	0.400	0.326
	low	**0.209**	0.253	0.245	0.259	0.285	0.243

Table 2 shows the result with the 1% validation set.[1] The proposed method outperformed the compared methods on all datasets except on *enron* and *genbase* with low-level noise. For *enron*, this is probably due to the small number of validation sets compared to the number of classes. In this situation, validation data tend to lack the relevant labels of the data to be estimated. For *genbase*, this is probably due to the fact that the average number of labels is 1.07. Indeed, it is known that this dataset is relatively easier to classify in the standard multi-label classification [6]. Compared to **PML-MD** and **Baseline**, the proposed **PML-VD** significantly improves the classification accuracy in most cases. **PML-NI** showed comparative performance on low-level noises even though this method does not assume the validation set. However, on high-level noises, the performances were negatively impacted by false-positive labels. Table 3 shows the results with the 3% validation set. The results are almost the same as that

[1] Due to the space limitation, we report the detail only with 1% and 3% validation sets.

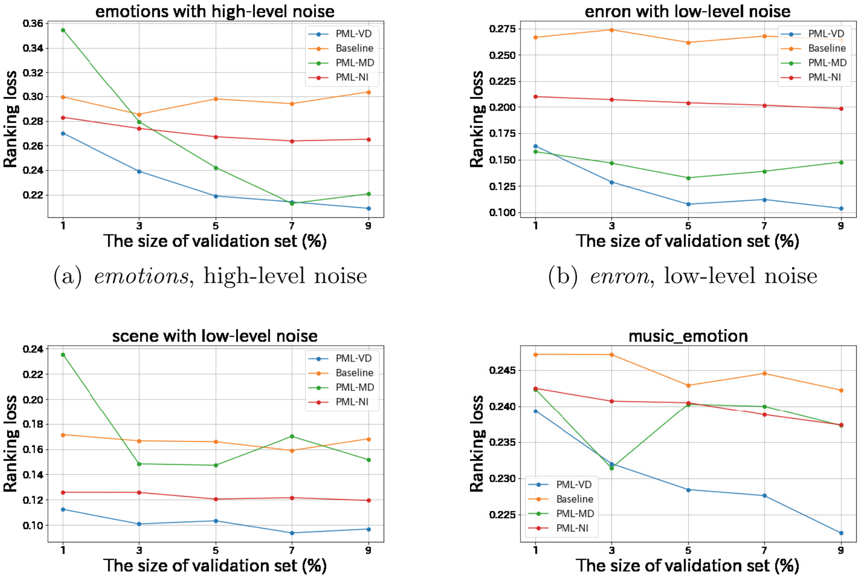
Table 3. Ranking loss on 3% validation set. The bold indicates the best method.

data	noise	**PML-VD**	**PML-MD**	**Baseline**	**PML-NI**	**PAR-MAP**	**PAR-VLS**
music_ emotion		0.232	**0.231**	0.247	0.241	0.363	0.266
music_ style		**0.133**	0.149	0.136	0.137	0.239	0.163
mirflickr		**0.063**	0.066	0.071	0.090	0.241	0.077
emotions	high	**0.239**	0.279	0.285	0.274	0.467	0.355
	low	**0.191**	0.240	0.228	0.217	0.470	0.215
enron	high	**0.135**	0.161	0.308	0.282	0.365	0.460
	low	**0.129**	0.147	0.274	0.207	0.319	0.283
CAL500	high	**0.243**	0.249	0.359	0.322	0.351	0.383
	low	**0.227**	0.245	0.289	0.251	0.269	0.343
scene	high	**0.145**	0.237	0.259	0.223	0.512	0.189
	low	**0.101**	0.148	0.167	0.126	0.472	0.117
genbase	high	**0.018**	0.230	0.049	0.021	0.518	0.247
	low	0.010	0.161	0.034	**0.006**	0.356	0.035
yeast	high	**0.197**	0.266	0.321	0.278	0.390	0.331
	low	**0.192**	0.259	0.247	0.260	0.285	0.247

with the 1% validation set. However, **PML-MD** slightly performed better than **PML-VD** on *music_ emotion* dataset. This implies that if the number of validation sets becomes large, it would be enough to estimate confidence without over-fitting. On the other hand, on **PML-NI**, **PAR-VLS**, and **PAR-MAP**, the improvements from the 1% validation set (on Table 2.) are limited compared to **PML-VD** and **PML-MD**. This implies that those methods cannot utilize accurately labeled data effectively.

Figure 2 (a) shows the results on *emotions* dataset with high-level noise. As seen, on **PML-MD** and **PML-VD**, the classification accuracy improves as the size of the validation set increases. On the other hand, the accuracies of **PML-NI** and **Baseline** which do not take validation set into account explicitly were steady. This shows the importance to use the validation set explicitly. In addition, **PML-VD** showed significant improvement against **PML-MD** when the validation set is small. This implies that **PML-MD** mitigates the negative effect of over-fitting. Figure 2 (b) shows the result on *enron* dataset with low-level noise. We can see **PML-VD** performed worse than **PML-MD** when the validation set is small (1%) but it improves as the size becomes larger. Figure 2 (c) shows the result on *scene* dataset with low-level noise. On this dataset, **PML-VD** showed steady performances with a varied size of the validation set. This is probably because the number of instances on *scene* dataset is relatively large and thus even it is enough to estimate the confidence with 1% validation set. On the other hand, **PML-MD** suffered from over-fitting. This also supports the proposed **PML-VD** mitigates the over-fitting issue. Figure 2 (d) shows the results on *music_ emotion*. In Table 3, **PML-MD** outperformed **PML-VD** on the 3% validation set. However, as seen in Fig. 2 (d) this is only on 3%. This is probably because the 3% validation set was chosen to fit the whole data. The selection of instances to remove false-positive labels is out-of-scope in this paper. However, this is one of our future work.

We analyzed the accuracy of the estimated confidence $\hat{p}$ on noisy set D to show how effective the first stage of the proposed **PML-VD** is. We measured Mean Squared Error between the estimated confidence $\hat{p}$ and true labels y. Figure 3 (a) shows the result of *music_ emotion* and Fig. 3 (b) shows the result of *scene* with high-level noise. As seen in both, **PML-VD** estimates confidence better than **PML-MD** and **Baseline** (candidate labels $\tilde{\boldsymbol{y}}$). This better confidence estimation brings better classification accuracy shown in Table 2 and 3.



(a) *emotions*, high-level noise

(b) *enron*, low-level noise

(c) *scene*, low-level noise

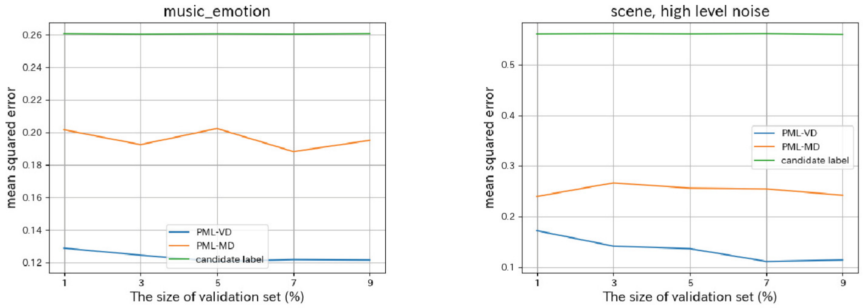
(d) *music_ emotion*

Fig. 2. Sensitivity analysis on the size of the validation set.

4.3 Parameter Analysis

We analyzed the relation between the noise and the parameter α on the ranking loss with fixing parameter $k = 9$.[2] Fig. 4 (a) and (b) show the result of the parameter analysis in the *scene* dataset with 1% and 5% validation set, respectively. As shown in Fig. 4 (a), with the 1% validation set, the ranking loss first appears to decrease as the parameter α (the weight on the noisy set) is increased. This indicates that the confidence estimation from the noisy set is effective to some

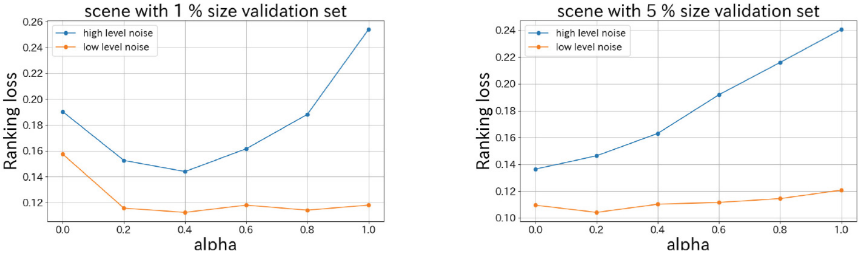
[2] PML-VD also has another parameter k but the effect is small and thus omitted due to the space limitation.



(a) *music_ emotion* (b) *scene*, high-level noise

Fig. 3. The accuracy of the estimated confidence $\hat{p}$ on the noisy set

extent. However, when the parameter α increases more than a certain value, the ranking loss starts to increase. This can be considered as noise labels affect when it is too much weighted, and the confidence estimation becomes inaccurate. In fact, this is especially remarkable for the high-level noise data. This shows the importance to estimate the confidence not from just one but both the validation set and the noisy set. On the other hand, in Fig. 4 (b), with the 5% validation set, the best accuracy was achieved with $\alpha = 0$ on high-level noise and with $\alpha = 0.2$ on low-level noise. This indicates that when the validation set is large enough, it is better to use only the validation set especially when the noise level is high. **PML-VD** can handle various settings by changing the parameter α depends on which sets are more reliable.



(a) *scene*, size of validation set is 1% (b) *scene*, size of validation set is 5%

Fig. 4. Sensitive analysis on the weight parameter α.

5 Conclusion

In this paper, we consider partial multi-label learning using a few accurately labeled data. In practice, it does not require a large cost to remove false-positive labels in just a few instances. However, all conventional PML methods except [10] cannot utilize the accurately labeled data since they do not assume that is available. Besides, an iterative learning approach in [10] could potentially lead to over-fitting due to the accumulation of errors. To mitigate this problem, we proposed a two-stage method called PML-VD. PML-VD takes a few accurately labeled data into account explicitly. In experiments on nine real and synthetic datasets, the proposed PML-VD showed its effectiveness compared to other conventional PML methods especially when the number of accurately labeled data is small and the noise on labels is large. In best cases, PML-VD improved the 7% accuracy in terms of ranking loss.

Acknowledgment. This work was partially supported by JSPS KAKENHI Grant Number 19H04128.

References

1. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
2. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: De Raedt, L., Siebes, A. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44794-6_4
3. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, pp. 39–43 (2008)
4. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Society for Industrial and Applied Mathematics (1995). https://doi.org/10.1137/1.9781611971217
5. Lyu, G., Feng, S., Li, Y.: Noisy label tolerance: a new perspective of partial multi-label learning. Inf. Sci. **543**, 454–466 (2021)
6. Read, J., Perez-Cruz, F.: Deep learning for multi-label classification. arXiv preprint: arXiv:1502.05988 (2014)
7. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. Mach. Learn. **39**(2), 135–168 (2000)
8. Xie, M.K., Huang, S.J.: Partial multi-label learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
9. Xie, M.K., Huang, S.J.: Partial multi-label learning with noisy label identification. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 3676–3687 (2021)
10. Xie, M.K., Sun, F., Huang, S.J.: Partial multi-label learning with meta disambiguation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1904–1912 (2021)
11. Zhang, M.L., Fang, J.P.: Partial multi-label learning via credible label elicitation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3587–3599 (2020)
12. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2013)
13. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine learning (ICML-03), pp. 912–919 (2003)

Shareable and Inheritable Incremental Compilation in iOOBN

Md Samiullah[1,2(✉)], Ann Nicholson[2], and David Albrecht[2]

[1] University of Dhaka, Dhaka, Bangladesh
samiullah@du.ac.bd
[2] Monash University, Melbourne, Australia
ann.nicholson@monash.edu, dwalbrecht724@gmail.com

Abstract. Object-oriented Bayesian networks (OOBNs) allow modellers to construct compositional and hierarchical models, using an inheritance hierarchy of classes ad subclasses, enabling reuse and supporting maintenance. Reasoning with both ordinary Bayesian networks (BNs) and OOBNs requires the important computational task of inference, the computing of new posterior probability distributions given a set of evidence. A widely used inference technique in ordinary BNs involves compiling the BN into a so-called junction tree (JT) before performing the inference; the compilation step is only performed when the model changes. In current OOBN software, the OOBN is first transformed into the underlying BN, so-called flattening, then the standard inference is performed. Researchers have proposed methods for incremental compilation of BNs, rather than recompiling from scratch for each network modification; these can apply to OOBNs also after flattening. Here, we propose a new incremental compilation technique that reuses existing compiled JTs of both embedded components and superclasses, and does not require flattening. We demonstrate through experimental analysis that this can reduce compilation time, and produces compact JTs that are cost-effective for inference.

Keywords: Graphical Models · OOBN · Incremental Compilation

1 Introduction

Bayesian networks (BNs) [5] are a powerful and widely used tool for reasoning under uncertainty. They can be built by automated learning if data is available, or using elicitation methods to capture expert knowledge when it is not. Especially when most or all of the model is built by hand, BN modelling methods don't scale up well; the resultant large complex BNs are difficult to visualise and hard for the domain experts and decision-makers to understand, reducing the acceptance and subsequent use of the model. Researchers have tried to address this issue by dividing the problem into subparts and then combining the BN models for the subproblems, and by re-using with some modifications of BN models previously built and validated for another application. These techniques include object-oriented BNs (OOBNs) [1,6], PRM and OOPRM, generalised decision-graphs, BN fragments, varieties combining probabilistic relational

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 91–103, 2024.
https://doi.org/10.1007/978-981-99-7022-3_8

models and objects, such as module networks, probabilistic relational models and plate models, multi-entity BNs (MEBNs), and template-based representations.

In this paper, we focus on a variant of OOBNs, that includes all the key concepts introduced by Koller and Pfeffer [6]: sub-parts of the overall model are represented in classes, which contained both nodes and objects, which are instances of other classes, giving a composite and hierarchical structure. These provide the advantages of OO software engineering concepts such as encapsulation, abstraction and information hiding, and support the building of large OOBNs in parallel by multiple modellers. OOBNs also provide modularity, which limits the scope of changes and reduces the chance of a model change introducing errors. The first implementation of OOBNs, without inheritance, was Hugin [7], a widely used commercial BN software tool. The research BN software UnBBayes [8] also includes OOBN functionality, also without inheritance, while the OOBN framework presented in [1] provides a limited form of inheritance. More recently, our iOOBN framework [13, 15] extends the Hugin OOBN implementation with a full treatment of inheritance and its associated OO concepts, namely object formation, instantiation, polymorphism, dynamic maintenance, and type checking.

Inference in BNs is the computation of new posterior probability distributions given evidence for a set of nodes. One widely used approach is to compile the BN into a so-called Junction Tree (JT) (e.g. [3]). To our knowledge, in Hugin and UnBBayes, to perform inference in OOBN, the network is first transformed into the underlying "flattened" ordinary BN, then a JT-based compilation is performed. Flores et al. [2] proposed "Incremental Compilation" (InC), to make ordinary BN inference more efficient by only re-compiling part of the network, with Bangsø et al. [1] proposing a similar incremental compilation for OOBNs, after flattening the network first.

In this paper, we present a new incremental inference algorithm (Sect. 3) for iOOBN that re-uses compiled structures constructed for either embedded classes, or when the new class is inheriting from a previously compiled class. Then we evaluated it extensively in Sect. 4 using a large synthetic dataset. Finally, we conclude our paper in Sect. 5 and outline plans for extending this work.

2 Background

A **Bayesian network (BN)** (following [5]) is a Directed Acyclic Graph (DAG) given by a 3-tuple $< V, E, \Pi >$, where (i) V = a set of **nodes** representing random variables, (ii) E = a set of directed **edges** representing the direct dependencies between nodes, with no directed cycles (iii) Π = a set of conditional probability distribution (CPD), one for each node. A node v_i is a **parent** of node v_j if there exist an edge $v_i \rightarrow v_j$. For each $v \in V$, $par(v) \subset V$ is the set of parent nodes of v, and the CPD $P(v|par(v))$ is a function Φ: $par(v) \cup \{v\} \rightarrow [0:1]$.

Formally, **Inference** in a BN is the process of calculating posterior probabilities of a set of variables X (where each variable is represented as a node v), given a set of evidence E and can be denoted as $P(x_i|E)$ for $x_i \in X$.

Next, we give the iOOBN definitions and terminologies required for the proposed new compilation algorithm; these follow and extend the OOBNs definitions and terminologies used in [5, 15] and implemented in Hugin BN. Note that while iOOBN is defined for Bayesian decision networks [5], incremental compilation applies only to the chance nodes representing random variables, so we limit the definitions given and do not consider decision and utility nodes. We also limit the treatment to discrete Bayesian networks, where all nodes have discrete state spaces.

An iOOBN **Concrete Class** C is a Directed Acyclic Graph (DAG) given by a 4-tuple $< V, O, E, \Pi >$, where V = a set of nodes representing random variables, E = a set of edges and Π = set of CPDs, one for each node, and O is a set of **objects**, representing instances of an iOOBN concrete class. We say that each $o \in O$ is **encapsulated** within C, the **encapsulating** class. An **instance** C_i^I is a replica or **instantiation** of a concrete class C_i with all the properties of that class.

Note that the iOOBN framework [15] has another type of class, an abstract class, for which some of the parameters Π are not fully defined. Since only instances of concrete classes can be compiled, and hence are the only classes used in the compilation algorithm, for the remainder of this paper, we refer to concrete classes as simply classes. Note also that once two iOOBN nodes have been joined by a referential edge, it implies that they represent the same random variable, which must be taken into account in any inference algorithm, including ours. The edges within an iOOBN must be such that it will "flatten out" to a valid BN, that is, a directed, acyclic graph.

An **iOOBN Subclass** $C' =< V', O', E', \Pi' >$ of a **Super** class $C = < V, O, E, \Pi >$ is a class that inherits the interface nodes (input nodes V_I and output nodes V_O) of C, implying that $V_I \subseteq V_{I'}$ and $V_O \subseteq V_{O'}$. Figure 1 shows an example iOOBN class having two input nodes (dashed ovals) namely A and S, two embedded objects (rectangles) and an output node N (double-lined oval). The embedded object on the left is an instance of the well-known Asia BN where V and S are also the names of its input nodes, X and D are its output nodes, and T, L, B, and F are its embedded nodes. The other embedded object is an instance of another class with A and S also being its input nodes, X and G being its output nodes, and C, L, and R being its embedded nodes. This class flattens into the BN given in Fig. 3 [P].

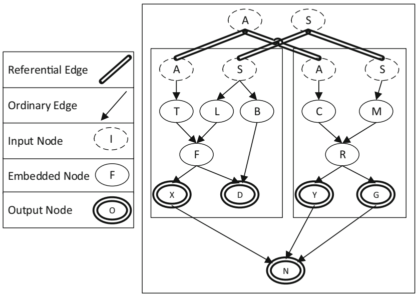


Fig. 1. Example iOOBN Class

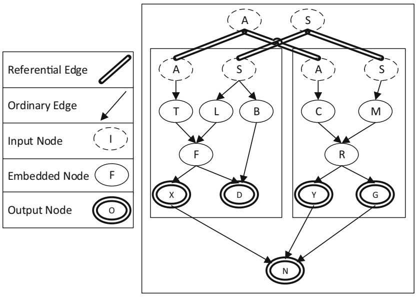


Fig. 2. Limitation of Flores's Inc. Compilation

A **Clique Graph** $CG = \{V, E\}$ of an iOOBN class C is a weighted undirected graph where V is a set of clique nodes, $V = \{Clq_1, Clq_2, ..., Clq_n\}$, with each clique containing nodes of C, and E is a set of edges, where each edge is a connection between a pair of clique nodes Clq_i and Clq_j and the weight of the connection is $|Clq_i \cap Clq_j| \geq 1$. $Clq_i \cap Clq_j$ is called the **separator set**. Associated with each clique node is a probability potential, which is a function of the variables in the clique, used later in inference. The product of all the potentials is the joint probability of all the variables in the clique graph.

A graph, denoted as JT, is a **Junction tree** (the basis of inference in BNs and the outcome of compilation) if it is a clique graph that (i) is a tree; and (ii) it has the running intersection property: for any pair of cliques, Clq_i, $Clq_j \in JT$, all the cliques in the path between Clq_i and Clq_j in the tree must contain $Clq_i \cap Clq_j$. A **Junction Forest**, JF, is a finite set of disjoint junction trees.

2.1 Inference

Inference, the most important purpose behind BN construction, has been extensively studied and explored in numerous pieces of research. One of the very widely used techniques of BN inference is “JT Based Inference” [3]. The main steps of this method are (i) moralization, (ii) triangulation, (iii) clique graph formation (where nodes are cliques and any two nodes will be connected by an edge if there are common items between them with weight equal to the number of common items), (iv) formation of JT/ Junction Forest (finding Maximum spanning tree of the clique graph, where cost function works on the weight of the edge), (v) message passing to propagate joint probabilities.

The incremental compilation (InC), proposed by Flores et al. [2], was motivated by the fact that all the operations in JT-based Inference, especially triangulation and clique finding, are computationally expensive [10]. InC is an MPS (Maximal Prime Subgraph) decomposition [12] based compilation technique where any modification to the BN does not require performing the above-mentioned steps and constructing JT from scratch. Instead, it constructs an MPS tree in parallel with JT construction during ordinary BN compilation. It keeps track of the changes done in the last BN structure and marks the affected parts of the MPS tree and the JT. Then the marked portion is re-triangulated and an intermediate JT for only the affected portion is constructed, which finally replaces the marked portion of the original JT. The InC method is particularly useful when the modification to the BN is minor and local, with expensive operations avoided for parts of the networks that are unchanged.

To our knowledge, there is no inference algorithm that works on the OOBN structure itself. In Hugin, an OOBN is flattened into an ordinary BN and any traditional exact or approximate inference technique may be applied. Any change, however minor, to the OOBN structure, generates full re-compilation, starting from flattening the new OOBN.

In an OOBN framework that supports inheritance, such as iOOBN (i.e. iOOBN [14, 15]), where any OOBN class can be derived from another class, then any change in the hierarchy generates a series of changes to all the subclasses below it in the inheritance hierarchy, so incremental compilation becomes even more important.

A potential solution to avoid the cost of repeated JT construction that utilises InC was proposed by Bangsø [1], with an implemented version described by Merten [9]. Changes in the OOBN classes are transformed into a series of equivalent changes of the corresponding flattened BN, then InC is applied. There are also some situations where, using this method, a large portion of the BN needs to be re-triangulated (because InC re-triangulates affected portions of the MPS tree); we give an example in Sect. 3.1.

Algorithm 1: SIIC

```
   Input: $C$: an iOOBN class
   Output: $JT_{new}$: A JT
1  begin
2      $C \leftarrow$ Preprocessing($C$)   /* Adding pseudo Ref. edges */
3      $< JF, RE > \leftarrow$ CreateJunctionForest($C$)
4      $Edges \leftarrow$ ConnectJunctionTrees(JF, RE)
5      $JT_{new} \leftarrow \phi$
6      foreach $edge\ E \in Edges$ do
7          $\{Clq_i, Clq_j\} \leftarrow$ getTerminalCliques($E$)
8          $JT_i \leftarrow$ JunctionTree($Clq_i$)
9          $JT_j \leftarrow$ JunctionTree($Clq_j$)
10         if $JT_i == JT_j$ then
11             /* connection between two cliques of the same JT */
12             $JT_{new} \leftarrow$ AddByMaintainingJTProperty ($JT_i$, $Clq_i$, $Clq_j$)
13         else
14             $JT_{new} \leftarrow$ JoinJTPair($JT_i$, $JT_j$, $Clq_i$, $Clq_j$)
15         $JT_{new} \leftarrow$ PostPruning($JT_{new}$)
16     $JT_{new} \leftarrow$ Thinning($JT_{new}$, 3) // Assuming minimum clique size = 3
17     $JT_{new} \leftarrow$ PostPruning($JT_{new}$)
18     return $JT_{new}$
```

Algorithm 2: Thinning

```
/*It performs a specialised operation, thinning, on the JT*/
Input: $JT$: a JT,
       $\tau$: Clique size threshold
Output: $JT$: a thinned JT
begin
    $CS \leftarrow$ stack of cliques in $JT$ of size $\geq \tau$
    while $CS$ is not empty do
        $C \leftarrow CS.pop()$
        $NC \leftarrow$ NeighborCliques($C$)
        $AOC \leftarrow$ AssociatedOriginalCliques($C$)
        $JT\_is\_thinner \leftarrow$ True
        foreach Partition $P$ of $AOC$ into $|NC|$ parts do
            $NewCliques \leftarrow \phi$
            foreach Part of $Cliques$ of $P$ do
                $NP \leftarrow$ MergeCliques($Cliques$)
                if $|NP| == |C|$ then
                    $JT\_is\_thinner \leftarrow$ False
                    break
                $NewCliques \leftarrow NewCliques \cup NP$
            if $JT\_is\_thinner$ then
                $JT \leftarrow$ Replace($JT$, $C$, $NewCliques$)
                push any clique in $NewCliques$ of size $\geq \tau$ onto $CS$
                break
    return $JT$
```

3 SIIC Compilation Algorithm

In this section, we present our proposed "Shareable Inheritable Incremental Compilation" (SII Compilation) algorithm that constructs the junction tree (JT) for an iOOBN, by re-using previously constructed JTs (of embedded objects and of superclasses) without flattening it into ordinary BN. The algorithm can significantly reduce the amount of recompilation required when a class in an iOOBN class hierarchy is modified.

The proposed SIIC algorithm takes an iOOBN class C as input, performs some preprocessing, retrieves any previously constructed JTs for its superclass (if there is one) and for any of its embedded objects, constructs any new JTs required, giving a Junction Forest (JF), then connects the JTs in JF to form a single resultant JT for C. The algorithm (see Algorithm 1[1] has four main stages: (i) Pre-processing (line 2) (ii) Creating a junction forest, including previously compiled JTs (line 3), (iii) Construction of the JT (lines 5–15), and (iv) Post-processing (lines 16–17).

In the **Pre-processing stage**, for each ordinary edge from an output node in an embedded object to an embedded node in the encapsulating class, we introduce a copy

[1] The SIIC code along with the iOOBN implementation is available on GitHub https://github.com/MdSamiullah/iOOBNFinal_v1.git.

of the output node together with a referential edge between the original output node and its copy (see Fig. 3 [B]). We will refer to these referential edges as *pseudo-referential edges*, as in practice these edges do not have to ever exist; this step is only added to simplify the explanation of the algorithm.

The **second stage** requires the recursive generation of the junction forest (via Algorithm 3). First, we retrieve (if previously compiled) or recursively create a JF from the superclass. We then (Algorithm 3, Line 10) identify components (nodes and edges) of the class being compiled that were not present in the superclass; this will be all of them if no superclass exists. The referential edges and pseudo-referential edges are then removed (Lines 11–13), as well as any embedded object, before these components are used to create new JT(s) (Line 14); this can be done using any JT-based BN inference, since there are no embedded objects. This is the base step of the recursion.

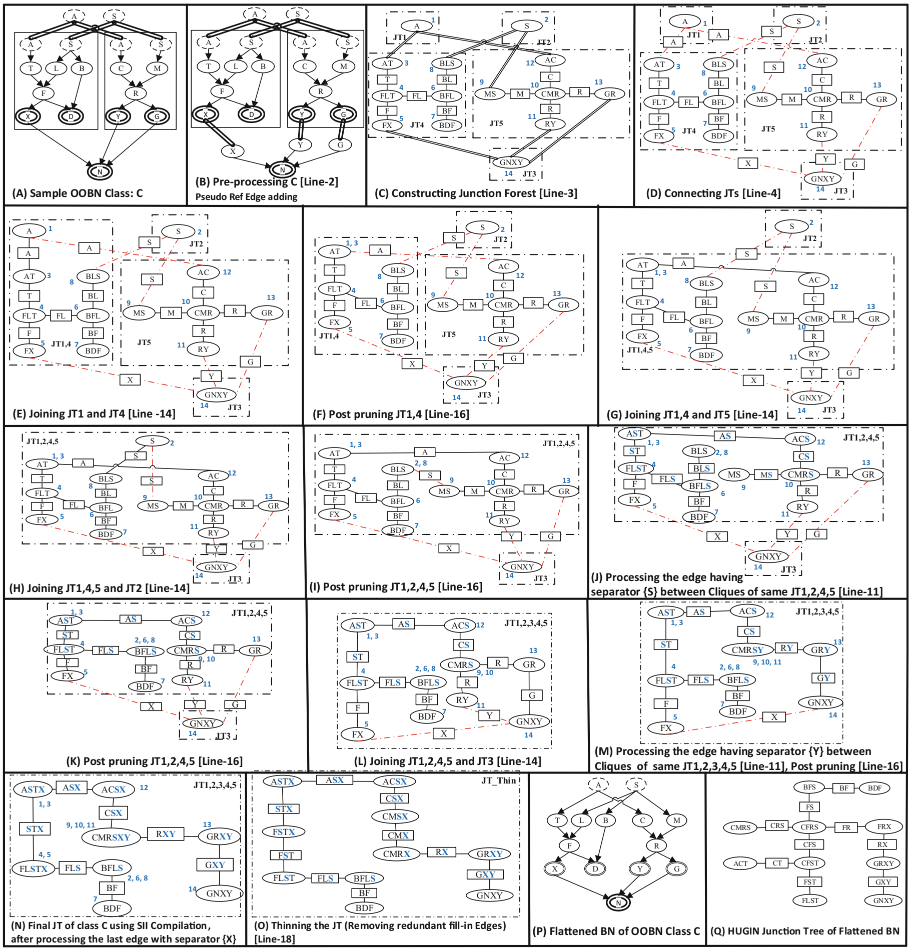
Algorithm 3: CreateJunctionForest

```
1  /* It creates a set of JTs */
   Input: $C$: an iOOBN class
   Output: $JF$: A set of JTs
           $RE$: A set of referential edges
2  begin
3      $C_{sup} \leftarrow$ superClass($C$)
4      if $C_{sup}$ exists then
5          if $C_{sup}$ is Compiled then
6              $JF \leftarrow$ getJunctionForest($C_{sup}$)
7          else
8              $JF \leftarrow$ CreateJunctionForest($C_{sup}$)
9      /*If any class C is extended from $C_{sup}$ with additional components in C then only
       the Junction Forest for $C - C_{sup}$ needs to be formed*/
10     $C \leftarrow$ removeComponents($C$, $C_{sup}$)// Remove inherited components from $C$
11     $RE \leftarrow$ popOutReferentialEdges($C$)
12     $Obj \leftarrow$ popOutInstanceNodes($C$)
13     /*TraditionalJTConstruction( ) returns a set of JTs using traditional JTBased
       inference approach*/
14     $JF \leftarrow JF \cup$ TraditionalJTConstruction($C$)
15     /*Due to the deletion of instance nodes and Referential edges, the remaining C will
       return a Junction Forest with at least one JT.*/
16     foreach instance $O \in Obj$ do
17         /* Get/create the JF for the corresponding classes of the instances */
18         if $O$.class is Compiled then
19             /* a compiled class will have its JT constructed */
20             $JF \leftarrow JF \cup$ getJunctionForest($O$.class)
21         else
22             $JF \leftarrow JF \cup$ CreateJunctionForest($O$.class)
23     return $JF$, $RE$
```

In the **third stage**, the JTs are then linked together into a single JT. During this process, the latest clique graph (JT_{new}) is regularly pruned, which involves removing unnecessary cliques and separators. Finally, in the **Post-processing stage**, we do some thinning (Algorithm 2), which involves splitting large cliques up into a path of smaller connected cliques, using information about the cliques they were constructed from; this thinning step is a simple linear checking and removal of redundant fill-in edges similar to the recursive thinning proposed in [5]. We then do a final pruning step.

Theorem 1. *SII compilation (Algorithm 1) generates a valid junction tree.*

Proof. We omit the proof for reasons of space and refer the interested reader to [13, Theorem 1, Page 125, Sect. 4.3, Chap. 4].



(A) Sample OOBN Class: C

(B) Pre-processing C [Line-2]
Pseudo Ref Edge adding

(C) Constructing Junction Forest [Line-3]

(D) Connecting JTs [Line-4]

(E) Joining JT1 and JT4 [Line -14]

(F) Post pruning JT1,4 [Line-16]

(G) Joining JT1,4 and JT5 [Line-14]

(H) Joining JT1,4,5 and JT2 [Line-14]

(I) Post pruning JT1,2,4,5 [Line-16]

(J) Processing the edge having separator {S} between Cliques of same JT1,2,4,5 [Line-11]

(K) Post pruning JT1,2,4,5 [Line-16]

(L) Joining JT1,2,4,5 and JT3 [Line-14]

(M) Processing the edge having separator {Y} between Cliques of same JT1,2,3,4,5 [Line-11], Post pruning [Line-16]

(N) Final JT of class C using SII Compilation, after processing the last edge with separator {X}

(O) Thinning the JT (Removing redundant fill-in Edges) [Line-18]

(P) Flattened BN of OOBN Class C

(Q) HUGIN Junction Tree of Flattened BN

Fig. 3. Example: SIIC in iOOBN [Line numbers of Algorithm 1 are shown]

Worked Example: Figure 3 shows how Algorithm 1 constructs a JT for the example OOBN in Fig. 3[A]. Note that this example only shows the re-use of the JT from the embedded object JTs, but does not show the re-use of a JT from a superclass. Figure 3[B] shows the result after pre-processing where duplicate names have been changed (L changed to M, and X to Y in the right embedded object), and pseudo-referential links are added to the copies of the embedded output nodes S, Y, and G that are parents of N. In Fig. 3[C], the JF has been formed using previously compiled JTs, JT_4 and JT_5, as well as the derived JTs, namely JT_1, JT_2 and JT_3, and connected into a JT via referential edges (shown with double lines); e.g. JT_4 and JT_3 are connected via a referential edge between clique nodes FX and GNXY. Note that each clique is given a unique index, shown in blue. Figure 3[D] shows the networks after these referential edges have been converted to edges in the clique graph (red-coloured dashed lines), indicating connections between the JTs, with the separators also shown; e.g., X is the separator on the red-coloured edge connecting JT_4 and JT_3. Figure 3[E] shows the result after joining JT_1 and JT_4. Next, post-pruning removes clique node 1 containing only variable A, shown in Fig. 3[F]; note that clique AT is now labeled "1,3", indicating it was created by the merger of cliques 1 (A) and 3 (AT). Figure 3[G] shows the result after joining $JT_{1,4}$ and JT_5; this step was straightforward, with the edge between AT and AC (with separator A) remaining and no other changes or post-processing are required. Similarly, Fig. 3[H] shows the formation of $JT_{1,2,4,5}$, then post pruning merging clique 2 (S) with clique 8 (BLS) in Fig. 3[I]. Next, the removal of the connection between BLS and MS by adding S (shown in blue in Fig. 3[K]) where required to cliques along another path between BLS and MS, to preserve the running intersection property, giving Fig. 3[K]. Figure 3[L] shows the structure after joining all five original JTs in the JF, $JT_{1,2,3,4,5}$, while Fig. 3[M] shows the result after removing the connection from RY to GNXY (post-pruning step), which required adding Y to two other cliques; at this point, there is only one unresolved connection from the original referential edges remaining (coloured red), that from FX to GNXY. Figure 3[N] shows the JT after merging cliques 4 and 5, and resolving that final edge, with the associated addition of X to cliques 13, 9-10-11, 1-3 and the new 4-5, and another post-pruning.

The last remaining step is the thinning, which uses the information about the original cliques that were stored throughout the JT combination steps (via the blue numbers associated with each clique). The thinning takes large cliques (using a clique size threshold) and iteratively splits them. For example, the CMRSXY clique is removed, first replaced by CMSX and CMRXY, then by CMSX, CMRX, RXY, and finally by CMSC and CMRX (after the final post-pruning of the unnecessary RXY); Fig. 3[O] shows the final resultant JT.

For comparison purposes, we show the flattened ordinary BN for the example OOBN class C (Fig. 3[P]) and the Hugin generated JT (flattening and recompiling from scratch) in Fig. 3[Q]. There are some similarities, e.g. the two cliques (GRXY and GNXY) at the right end of our JT, with the clique at the left end (BDR) also being a leaf in the Hugin JT. Overall, however, the JTs are quite different, with the SII Compilation JT tending to have larger cliques; we discuss this further in the following subsection.

3.1 Efficiency Analysis

The efficiency of the proposed technique can be evaluated from two perspectives: how much compilation time (i.e., time complexity of the JT construction) is reduced, and the efficiency of the resultant JT for subsequent inference.

1) Time complexity of JT Construction: SII Compilation allows re-use of the JT of its superclass, if it exists, and when the modification from superclass to subclass does not involve the removal of any of the superclass nodes and edges, as well as previously compiled JTs from embedded objects. At the base recursion of the algorithm, new JTs are compiled on a standard BN using any JT compilation method. However, it is not possible to give performance guarantees about the reduction of compilation computation because that will depend on the specific structure of the class being compiled and the structures of any embedded objects. Naturally, in the extreme case, if there are no superclasses or embedded classes (i.e. if it is not an iOOBN) SII Compilation will incur computation overheads without any reduction in compilation computation time.

The widely used traditional flattening-based approach always starts from scratch and does not allow the reuse of existing outcomes at all. Its primary computation involves flattening of the OOBN as well as the standard triangulation and JT construction cost. Though flattening is a linear time operation, with hierarchies of embedded classes, it may introduce significant complexities to the computation [10]. The JT construction cost is polynomial in the number of cliques in the clique graph, the same as Prim's Minimum Spanning Tree construction cost. Minimal triangulation is an NP-Hard problem, though heuristic-based suboptimal triangulation requires polynomial time to the number of variables in the Bayesian network.

As described in Sect. 2.1, the incremental compilation (InC) approach allows reusing existing compiled structures, however, it still re-triangulates a portion of the existing structure as well as incurring extra computational and storage burden for the MPSD structure maintained in parallel with the JT. We have also observed that it may encounter scenarios where the whole network structure or a large portion of the structure needs to be re-triangulated, and therefore leads to similar computation as the traditional flattening-based approach. For example, suppose we are using InC to add an edge $X \rightarrow N$ to the network shown in Fig. 2. The figure also contains the JT for the initial network with the affected and marked JT segment due to the addition of the edge. The incremental compilation approach will affect 8 cliques out of 10 of the JT, meaning almost the whole network needs to be re-triangulated and the modified portion needs to be joined, to get the resultant structure. However, in our SIIC algorithm, a simple merging of JTs will be enough and that is a straight-forward operation.

2) Efficiency of the resultant JT: The structure of the resultant JT has implications for the subsequent inference computation time. A measure to roughly quantify this based on a message passing inference algorithm is the so-called JT cost (as proposed by Kanazawa [4]): $JT_{cost} = \sum_{C_i \in \{C_1,...,C_n\}} \left(K_i \prod_{X \in C_i} |\Omega_X| \right)$, where $C_1, ..., C_n$ represent the cliques in the JT, K_i denotes the sum of the number of parent and child cliques of C_i in JT (reflecting the arity of the JT), and Ω_X is the state space of node X in clique C_i. The JT cost, therefore, depends on the structure of the JT, as well as the size of the state spaces of the nodes of the OOBN class. For our example OOBN class C, if all the nodes are binary, The JT in Fig. 3[O] has 8 cliques with four variables (2^4 combinations) and two neighbours, 1 clique with 3 variables (2^3 combinations) and one neighbour, and 1 clique with 4 variables and 1 neighbour. This produces JT_{cost} = 280. The cost of the Hugin-generated JT is 240.

Although our JT cost is a little higher than the JT cost of Hugin, our produced JT is binary (a JT where no clique node has more than three connections). In the Shenoy-Shafer architecture, it is proved that a binary JT is more efficient than a non-binary JT, however a theoretical result about the arity of the resultant JT given the arity of the pre-compiled JTs remains an area for further research.

The overlapping factors influencing the performance of the compilation algorithms make it difficult to provide any performance guarantee. Hence, in the following section, empirical analyses are conducted on synthetic OOBN classes with various parameters and their different combinations.

Table 1. Experimentation parameters and terms

Parameters	Terms	Ranges (Values)	Parameters	Terms	Ranges (Values)
Num. of Nodes	NON	5, 10, 15, 20, 25, 30, 50	#Fold per configuration	Folds	5
Num. of States	NOS	2, 3, 4, 5	#Repeated Runs	Runs	4
Num. of Parents	NOP	2, 3, 4, 5	JT Exist?	SIIC	No
Num. of Foreign Classes	NOC	0, 1, 2, 3		SIIC#	Yes
Num. of Objects	NOO	1, 2, 3, 4	Average NOP	NOPAvg	–

Table 2. SIIC vs SIIC# vs Hugin (Runtime)

Breakdown	Hugin	SIIC	SIIC#	Hugin-SIIC	Hugin-SIIC#
Min	3.44	3.44	0	−3.15	−2.19
1st Qu	4.14	4.36	2.71	−0.38	**1.04**
Median	4.36	4.6	2.77	−0.13	**1.44**
Mean	4.87	4.78	3.21	**0.09**	**1.67**
3rd Qu	4.94	5.05	3.83	**0.19**	**2.11**
Max	10.59	9.98	7.16	**5.29**	**7.33**

4 Experimental Analysis

We next analyze and compare the performance of two versions of the proposed incremental algorithm, SIIC and SIIC# to Hugin. SIIC# is the same as SIIC except it uses pre-compiled JTs of all embedded objects and superclasses; in a sense, it provides an upper bound for the compilation time savings available to SIIC.

We compare the performance of the compilation algorithms in terms of (i) the runtime, and (ii) the cost of the JTs, JT_{cost} (see above). The compilation times for all 3 algorithms are computed with a PC (Intel(R) Core(TM) i5-8259U CPU, 8 GB RAM). Due to the unavailability of real-life iOOBN class repositories, the analysis was performed on a repository of synthetic OOBN classes that were generated using a range of values for parameters such as the number of nodes, states, parents, foreign classes (classes that are used to create embedded objects), embedded objects (see Table 1), in combination giving 1456 configurations ($7 \times 4 \times 4 + 7 \times 4 \times 4 \times 3 \times 4$). Five different OOBN models were generated for each configuration, producing 7280 OOBNs.

To produce synthetic OOBN classes for the experimentation, we first produce a set of BNs using the parameters (see the 2nd column of Table 1). Then we convert them into OOBN classes using a simple heuristic that identifies potential interface nodes from the set of nodes of a BN: (1) all nodes in the BN with no parent nodes (‘root’ nodes) are potential input nodes, and (2) all the nodes in the BN with no child nodes (‘leaf’ nodes) form the set of potential output nodes in the OOBN class.

We also look at the comparative performance of the algorithms in terms of the complexity of the OOBN, using the number of parameters in the OOBN as a measure of complexity proposed in [11]. For each OOBN, we ran the compilation algorithms 4 times, to reduce the fluctuation in run time, giving 29120 runs. We found that many of the generated OOBNs were too large and complex for the compilation algorithms to handle; they ran out of memory and didn’t produce a valid JT. For the analysis, we use only the results for OOBNs where all the algorithms produced JTs in 2 or more of the runs. Moreover, to deal with unusual behaviour of the values, e.g., scattered, some of the outliers were removed; this left us with 11043 runs across 3515 unique OOBNs.

Unsurprisingly, all three algorithms' running time increase as the size and complexity of the OOBN increases. We refer the interested reader to the thesis [13, Sec 4.6, Chap. 4] for a full set of results. However, a summary of the run-time comparison of the algorithms is given in Table 2. The 1st column of the table shows the distribution of the complexities of the input OOBNs, using the complexity measure M given above, in four quartiles (i.e. a tabulated version of results often shown as box plots). The next 3 columns show the run times for each algorithm, and then the differences between Hugin and SIIC and SIIC#, with bold indicating where SIIC or SIIC# performs better. It can be seen that SIIC performs better than Hugin in terms of runtime for the OOBNs in the upper half in terms of complexity. SIIC# performs better in all except the min case, although of course, this depends on pre-compiled JTs being available.

Table 3. Breakdown of failed runs (F=Fail, C=Complete)

Hugin	SIIC	SIIC#	**Count**	%
C	C	C	14060	48.28
F	C	C	7161	24.59
F	F	C	2908	9.99
F	F	F	4991	17.14
Total			29120	100

Table 4. t-test statistics of experimentation

Algo 1	Algo 2	Algo 1 wins		Algo 1 loses		Tied	
		cnt	%	cnt	%	cnt	%
Hugin	SIIC	312	8.88	65	1.84	3138	89.28
Hugin	SIIC#	4	0.11	762	21.68	2725	78.21
SIIC	SIIC#	0	0.00	1151	32.75	2340	67.26

Table 5. JT cost summary. **Bold** is better.

Breakdown	Hugin	SIIC	SIIC-Hugin
Min	2.77	2.77	−**5.60**
1st Qu	9.57	11.62	0.79
Median	12.89	16.07	2.35
Mean	13.02	16.63	3.62
3rd Qu	16.25	20.76	5.10
Max	21.89	49.67	31.23

Table 3 shows the number and percentage of cases in which the algorithms ran or failed. In 51% of cases, Hugin failed to produce any JT, and SIIC and SIIC# also failed in some runs because they call the Hugin compilation on a component (they never failed when Hugin completed). Unsurprisingly, SIIC# has the least number of failed runs.

Table 4 compares the algorithms (Hugin, SIIC, SIIC#) on their relative performance against each other and shows the number and percentage of wins, losses and tied outcomes for each. In order to make the comparisons statistically significant, the algorithms were compared using the paired t-test for the four runs of each of the trialled and distinct networks. The detailed comparison of how many times Hugin wins and loses against SIIC and SIIC#, how many times the algorithms tie, and how many times SIIC# outperforms SIIC, are listed. As mentioned, in the table, where it is clear that even if the 51% of times when Hugin failed are ignored, SIIC# outperformed both Hugin and SIIC. The percentage of cases in which Hugin outperformed SIIC is low, at 8.88%.

Next, we compare SIIC to Hugin compilation in terms of the cost of the JTs they produce; these results are given in Table 5. Note that we do not give separate results for SIIC#, as its JT is always identical to that of SIIC. Clearly the JT$_{cost}$ of SIIC is higher than the JT$_{cost}$ of Hugin, due to the SIIC JT's clique sizes being larger, as we saw above for our running example. So the improvement in compilation runtime SIIC obtained through the re-use of previously compiled components does come with a slight trade-off in terms of the complexity of the resultant JT.

5 Conclusions and Future Work

We have proposed a new incremental compilation algorithm, SII compilation, for OOBNs, that unlike previous methods does not require transforming the OOBN into its underlying BN. The SII method incorporates two kinds of re-use: (1) when compiling a subclass, re-use the JT of its superclass; and (2) when compiling a class with embedded objects, re-use the JTs from those objects. We have proven [13] that the algorithm produces a valid JT. The constructed JT from our algorithm may, and in examples tends to, contain larger cliques in comparison to the cliques of existing approaches. Nevertheless, after the final thinning step, the resultant JT can be more compact, using a JT-cost measure that captures the complexity of inference on that JT. We conducted an empirical analysis across a range of synthetic classes, exploring what benefits can be achieved in practice. Results showed that, as expected, compilation time and JT cost increase with the increase of the size and complexity of the OOBN (and underlying BN). In most of the cases, in terms of compilation runtime, SIIC outperforms Hugin's compilation (which flattens the OOBN into a BN and then compiles the BN), and SIIC# outperforms Hugin in all cases, particularly when the OOBN has embedded objects. In terms of JT-cost, Hugin's JTs are far better than non-thinned JTs constructed by SIIC (or SIIC#). Moreover, with the success of compilation also being a performance factor, both SIIC# and SIIC successfully generated JTs for iOOBN models that the Hugin inference could not handle. We plan to explore whether we can further optimise the algorithm to reduce the clique sizes, and also investigate whether there are theoretical results regarding the arity of the resultant JT.

References

1. Bangsø, O., Flores, M.J., Jensen, F.V.: Plug & Play object-oriented Bayesian networks. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.-L. (eds.) CAEPIA/TTIA -2003. LNCS (LNAI), vol. 3040, pp. 457–467. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25945-9_45
2. Flores, M.J., Gámez, J.A., Olesen, K.G.: Incremental compilation of Bayesian networks. In: Proceedings of 19th International Conference of Uncertainty in Artificial Intelligence UAI '03, pp. 233–240 (2003)
3. Jensen, F.V., Lauritzen, S.L., Olesen, K.G.: Bayesian updating in causal probabilistic networks by local computations. Comput. Stat. Q. **4**, 269–282 (1990)
4. Kanazawa, K.: Probability, time, and action. Ph.D. thesis, PhD thesis, Brown University, Providence, RI (1992)
5. Kjærulff, U.B., Madsen, A.L.: Bayesian networks and Influence Diagrams, vol. 200, p. 114. Springer, Cham (2008)
6. Koller, D., Pfeffer, A.: Object-oriented Bayesian networks. In: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI), USA, 1997, pp. 302–313 (1997)
7. Madsen, A.L., Lang, M., Kjærulff, U.B., Jensen, F.: The Hugin tool for learning Bayesian networks. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 594–605. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45062-7_49
8. Matsumoto, S., et al.: UnBBayes: a Java framework for probabilistic models in AI. Java Academia Res., 34 (2011)
9. Merten, C.: Incremental compilation of object-oriented Bayesian networks (2005)

10. Mezzini, M., Moscarini, M.: Simple algorithms for minimal triangulation of a graph and backward selection of a decomposable Markov network. Theory Comput. Sci. **411**(7–9), 958–966 (2010)
11. Nicholson, A., Flores, J.: Combining state and transition models with dynamic Bayesian networks. J. Ecol. Modell. **222**(3), 555–566 (2011)
12. Olesen, K.G., Madsen, A.L.: Maximal prime subgraph decomposition of Bayesian networks. IEEE Trans. Syst. Man Cybernet. Part B **32**(1), 21–31 (2002)
13. Samiullah, M.: iOOBN: an object-oriented Bayesian network modelling framework with inheritance. Ph.D. thesis, Faculty of IT, Monash University, Clayton, Vic, Australia (2020). available to view: https://tinyurl.com/268wt635
14. Samiullah, M., Albrecht, D., Nicholson, A.: Automated construction of an object-oriented Bayesian network (OOBN) class hierarchy. In: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE (2022)
15. Samiullah, M., Hoang, T.X., Albrecht, D., Nicholson, A., Korb, K.: iOOBN: a Bayesian network modelling tool using object oriented bayesian networks with inheritance. In: Proceedings of 29th ICTAI, BOSTON, MA, USA, 6–8 November pp. 1218–1225 (2017)

A Dynamic Pricing Strategy in Divided Regions for Ride-Hailing

Bing Shi[1,2(✉)], Yan Lu[1], and Zhi Cao[1]

[1] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China
{bingshi,yanlu_}@whut.edu.cn

[2] Shenzhen Research Institute of Wuhan University of Technology, Shenzhen 518000, China

Abstract. Nowadays, ride-hailing services play a significant role in daily transportation. In the ride-hailing system, the temporal and spatial distribution of demand and supply in different regions is different. Therefore, it is necessary to differentiate pricing for regions based on demand and supply. Instead of setting discriminatory prices by simply dividing the whole area into some fixed regions, which failed to take into account the demand and supply dynamics over time, we developed a dynamic region-division based pricing strategy according to demand and supply in different regions, with the goal of maximizing the platform's long-term profit. Furthermore, we perform comprehensive experiments on a real-world dataset to demonstrate the effectiveness of the proposed algorithm. The experimental results indicate that our algorithm can outperform other typical benchmark approaches.

Keywords: Ride-hailing · Demand and Supply · Dynamic Pricing

1 Introduction

Ride-hailing services have played a crucial role in people's daily transportation. In the ride-hailing system, passengers will choose the ride-hailing platform based on the requested riding price. Therefore, how to set the price for orders is a key issue. There exist some works which determine prices based on demand and supply, such as Lyft's Peak-hour Pricing, Uber's Surge Pricing, and so on [3,7]. However, these works consider dynamic pricing based on demand and supply over the whole area. In fact, different regions in the same city may have different demand and supply at the same time. At this moment, the platform may need to divide the whole city into several regions and set discriminatory prices for each region according to its demand and supply, such as [5,6]. However, these works are usually based on fixed region-division, i.e., the region size and shape do not change over time. In fact, demand and supply in the same region may change dynamically over time. In this situation, the platform may need to dynamically divide regions, and then develop an effective pricing strategy to determine prices for each divided region according to demand and supply. In this paper, we aim

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 104–110, 2024.
https://doi.org/10.1007/978-981-99-7022-3_9

to analyze how to dynamically set prices for orders based on the dynamic region-division, aiming to maximize the platform's long-term profit.

Specifically, considering that it is difficult to divide the entire area into some irregular regions, we first divide the whole area into a number of rectangular zones that do not overlap with each other. Intuitively, the platform should set the same or similar prices for zones with similar demand and supply and nearby locations. If not, passengers may leave the platform since they will feel unfair when they are charged discriminatorily and thus damage the long-term profit. Therefore, we design a dynamic region-clustering algorithm (**DRC**) by combining Deep Q Network (DQN) and K-Means algorithms to cluster zones with similar demand and supply and nearby locations into one region. Furthermore, we propose an adaptive multi-region dynamic pricing algorithm (**AMRDP**), which maximizes the platform's long-term profit based on the states of demand and supply in different regions. Finally, we run comprehensive experiments based on a real-world dataset. The results indicate that the platform's profit is increased under different pricing algorithms combined with **DRC**. Furthermore, the combination of **AMRDP** with **DRC** can bring higher profits, serve more orders, and have a higher service rate than typical benchmark approaches.

The remaining sections of this paper are structured as follows: We introduce the settings in Sect. 2, present the algorithm in Sect. 3, perform experimental analysis in Sect. 4, and summarize the paper in Sect. 5.

2 Basic Settings

In this section, we introduce the basic settings and give the problem formulation. The whole time period is divided into a group of time steps, denoted as $t \in \{1, 2, \cdots, T\}$. Since it is difficult to divide the area into irregular regions, similar to [8], we first divide the whole area into a set of non-overlapped rectangular zones, denoted as $g \in \{g_1, ..., g_N\}$, and then can cluster these rectangular zones into regions based on the demand and supply of each zone. Specifically, we use $\{\{g_1^1, g_1^2, \dots g_1^{c_1}\}, \{g_2^1, g_2^2, \dots g_2^{c_2}\}, \dots, \{g_{m_t}^1, g_{m_t}^2, \dots g_{m_t}^{c_{m_t}}\}\}$ to represent the division result when the whole area is divided into m_t regions at time step t, where $\{g_i^1, g_i^2, \dots g_i^{c_i}\}$ is the i-th region, which consists of c_i zones.

Definition 1 (Order). *Order $o_i \in \mathcal{O}$ is denoted as $o_i = \left(r_i, price_{r_i}, t_{r_i}^m\right)$, where $r_i = \left(l_{r_i}^e, l_{r_i}^s, t_{r_i}, g_{r_i}, val_{r_i}, f_{r_i}\right)$ is the riding demand, $l_{r_i}^e$ and $l_{r_i}^s$ are the drop-off and pick-up locations respectively, t_{r_i} represents the time when the r_i is submitted, g_{r_i} is the pick-up zone, val_{r_i} is the highest unit price per kilometer the passenger with r_i is willing to accept for the service, f_{r_i} is the status of r_i, $t_{r_i}^m$ is the highest waiting time and $price_{r_i}$ is the order's price.*

Specifically, val_{r_i} is private to the passenger and is unknown to the platform, and we assume it is independently and identically drawn from a uniform distribution within $[p_{\min}, p_{\max}]$[4].

Definition 2 (Vehicle). *Vehicle* $v_i \in \mathcal{V}$ *is denoted as* $v_i = (g_{v_i}, l_{v_i}, d_{v_i}, s_{v_i})$, *where* g_{v_i} *and* l_{v_i} *are the current zone and location of* v_i *respectively,* d_{v_i} *denotes the travel cost of the vehicle, and* s_{v_i} *is the vehicle status.*

We make the assumption that the vehicles belong to the platform and only serve orders within the region since the service range of the vehicles is limited.

Definition 3 (Platform's long-term profit). *The long-term profit of the platform is calculated as the total amount paid by all served passengers minus the overall cost of vehicles throughout the whole period.*

$$EP = \sum_{\mathcal{T}=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M_{\mathcal{T}}^{i}} G\left(p_{g_{r_i}}^{\mathcal{T}}\right) \mathbb{I}\left(f_{r_j} = 2\right) \left(p_{g_{r_i}}^{\mathcal{T}} \times \text{dis}\left(l_{r_j}^{s}, l_{r_j}^{e}\right) - C_{r_j}\right) \tag{1}$$

where N *and* T *are the number of zones and time steps respectively,* C_{r_j} *denotes the travel cost associated with serving riding demand* r_j*,* $M_{\mathcal{T}}^{i}$ *and* $p_{g_i}^{\mathcal{T}}$ *denotes the number of riding demands and unit price in zone* i *at time step* $\mathcal{T}$ *respectively,* $f_{r_j} = 2$ *means that an idle vehicle has been arranged to serve the passenger.*

Definition 4 (Dynamic Region-Division and Pricing Problem). *Given the riding demands* R_t *and idle vehicles* $\mathcal{V}_t$ *for the current time step, the platform should make a division decision according to the demand and supply of the current time step* $\sigma_t = \{\{g_1^1, g_1^2, \cdots g_1^{c_1}\}, \{g_2^1, g_2^2, \cdots g_2^{c_2}\}, \ldots, \{g_{m_t}^1, g_{m_t}^2, \cdots g_{m_t}^{c_{m_t}}\}\}$, *and then set the unit price* $P_t = \left(p_t^1, p_t^2, \ldots, p_t^{m_t}\right)$ *for each region in order to maximize the long-term profit* LP *over the whole time period.*

3 Dynamic Region-Division Based Pricing Strategy

3.1 Dynamic Region-Clustering Algorithm

As we mentioned previously, in order to prevent unfair treatment on passengers caused by discriminatory pricing in zones with similar demand and supply, we need to cluster these zones with similar demand and supply and nearby locations into the same region. In this paper, we first determine the number of regions in each time step and then use K-Means to cluster zones. Since the platform needs to divide regions in each time step and the division and further demand and supply are affected by each other, the dynamic region-division problem is modeled as a Markov decision process (MDP). Therefore, we can use a deep reinforcement learning algorithm to address the problem. We provide a detailed description of MDP as follows, which is denoted as (S, A, P, r, γ).

State: $s_t = (v_t, c_t) \in S$, where v_t and c_t are the numbers of idle vehicles and riding demands in each zone at time step t respectively.

Action: $a_t = m_t \in A$, where m_t is the number of clusters(regions).

Reward: r_t denotes the profit of the platform at time step t, which is:

$$r_t = \sum_{i=1}^{N} \sum_{j=1}^{\mathcal{R}_i} G\left(p_{g_{r_j}}\right) \mathbb{I}\left(f_{r_j} = 2\right) \left(p_{g_{r_j}} \times \text{dis}\left(l_{r_j}^{s}, l_{r_j}^{e}\right) - C_{r_j}\right) \tag{2}$$

where $\mathcal{R}_i$ and C_{r_j} denote the number of riding demands in zone i and the travel cost of the vehicle serving the riding demand r_j respectively.

P and γ are the state transfer probability function and discount factor respectively. Specifically, the value of γ typically ranges from 0 to 1, and we set it at 0.9 to achieve a balance between short-term and long-term rewards.

We use Deep Q Network(DQN) to determine the number of regions since the action space of the platform to determine the region numbers is discrete. The dynamic region-clustering algorithm(**DRC**) is presented in Algorithm 1.

3.2 Adaptive Multi-Region Dynamic Pricing Algorithm

Note that the dynamic pricing will impact future demand and supply in different regions, which will further affect the dynamic pricing. Therefore, it is also a sequential decision problem, and thus we can model it as a MDP and apply deep reinforcement learning to solve it. Now we describe the MDP as follows.
State: $s'_t = (a_{t-1}, e_{t-1}, mc_{t-1}, v_t, c_t) \in S'$, where a_{t-1} denotes the average unit price in each region, mc_{t-1} denotes total number of orders that were successfully matched with idle vehicles in each region and e_{t-1} denotes the total profit in each region at the last time step. Since the whole area is dynamically divided, the number of regions is changing dynamically. We add some virtual regions to fill the original state information into a fixed dimension, and we set the state information of these virtual regions to 0.

Algorithm 1. Dynamic Region-Clustering Algorithm(**DRC**)

Input: Distribution of the riding demands and vehicles

Output: Dynamic region-division strategy τ

```
Initialize action-value function network $\mathcal{Q}_\theta$, target value network $\mathcal{Q}_{\theta^-}$, replay memory $\mathcal{D}$ and set the weight $\theta^- \leftarrow \theta$;
for $\kappa = 1$ to $K$ do
    Initialize the state $s_1$;
    for $t = 1$ to $T$ do
        Choose the action $a_t$ by the $\epsilon$-greedy strategy, that is, the number of regions;
        Use $K$-Means algorithm to cluster zones into $a_t$ regions;
        Set the unit price and then match the orders;
        Get the reward $r_t$ and transfer to the next state $s_{t+1}$;
        Store the state transition tuple $(s_t, a_t, r_t, s_{t+1})$ into $\mathcal{D}$;
        Randomly select a set of samples $(s_i, a_i, r_i, s_i)$ from $\mathcal{D}$ for training;
        Calculate $\mathcal{Y}_i = \begin{cases} r_i & \text{terminated in generation } i+1 \\ r_i + \max_{a'} \mathcal{Q}\left(s_{i+1}, a'; \theta^-\right) & \text{otherwise} \end{cases}$;
        Calculate the loss function of $N$ sequences: $L = \frac{1}{N}\sum_i \left(\mathcal{Y}_i - \mathcal{Q}\left(s_i, a_i; \theta\right)\right)^2$;
        Perform gradient descent on $L$: $\nabla\theta = \frac{2}{N}\sum_{i=1}^{N}\left(\mathcal{Y}_i - \mathcal{Q}\left(s_i, a_i; \theta\right)\right)\nabla\mathcal{Q}\left(s_i, a_i; \theta\right)$;
        Update network parameters: $\theta = \theta + \nabla\theta$ and update $\theta^- = \theta$ every $\mathcal{C}$ step;
    end for
end for
```

Action: $a'_t = \left(p_t^1, p_t^2, \dots p_t^{m_t}\right) \in A'$, where p_t^i represents the unit price for i-th region at time step t.
Reward: $r'_t\left(a_t, s_t\right) = \mu_1 \times CP_t + \mu_2 \times ratio$, where CP_t denotes the profit of the platform at time step t according to Eq. 2, the parameters μ_1 and μ_2 are the normalization coefficients of CP_t and $ratio$ respectively, and $ratio$ signifies the proportion of the actual number to the highest achievable number of the served orders. Note that in the reward function, we consider both the platform's profit and order service rate. In so doing, we can guarantee that the platform will not set an excessively high price, which may cause passengers to leave the platform. This can increase the passengers' participation rates and lead to higher profits for the platform. Specifically, we tried different parameter combinations in the experiments to balance the impacts of the profit and service rate on the reward and finally set them as 1/70 and 2 respectively, which ensures that the dynamic pricing algorithm can effectively balance profit and service rate. And the value of γ' is also set to 0.9.

Due to the action space for pricing is continuous, we employ deep deterministic policy gradient(DDPG) to design an adaptive multi-region dynamic pricing algorithm(**AMRDP**), as described in Algorithm 2. At different time steps, due to the changing number of regions, the input state dimensions are different. Therefore, when the platform observes the state information about each region, it will add some virtual regions to fill the original state into a fixed dimension, and the state information of these virtual regions is set to 0. At the end of each round, the platform collects information about passengers who have accepted the platform's price and idle vehicles, and then matches idle vehicles with orders. We model the order matching as a bipartite graph maximum weighted matching problem, and use Kuhn-Munkres [2] to match vehicles with passengers, with the aim of maximizing the platform's profit in the current time step.

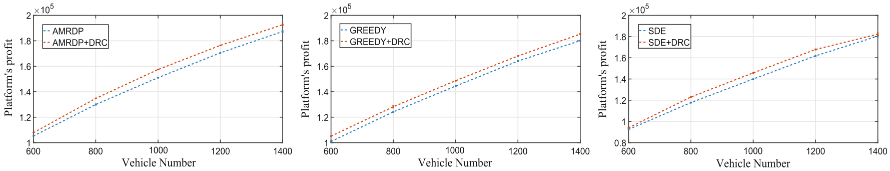
Algorithm 2. Adaptive Multi-Region Dynamic Pricing Algorithm(**AMRDP**)

```
Input: Distribution of the riding demands and vehicles
Output: Dynamic Pricing strategy $\pi$
Initialize Actor network $\mu\left(s \mid \theta^{\mu}\right)$, Critic network $\mathcal{Q}\left(s, a \mid \theta^{\mathcal{Q}}\right)$, replay memory $\mathcal{D}'$
  and set the weights of target network $\mathcal{Q}'$ and $\mu'$ as $\theta^{\mathcal{Q}'} \leftarrow \theta^{\mathcal{Q}}, \theta^{\mu'} \leftarrow \theta^{\mu}$;
for $\kappa = 1$ to $K$ do
    Initialize the state $s_1$ and noise function $\mathcal{N}$;
    for $t = 1$ to $T$ do
        Observe the state $s_t$ and pad it to a fixed dimension $s'_t$;
        Select the action $a_t = \mu\left(s'_t|\theta_{\mu}\right) + \mathcal{N}_t$ based on $s'_t$ and then executes it.
        Matches the orders and get the reward $r_t$ and move to next state $s_{t+1}$;
        Pad $s_{t+1}$ to the fixed dimension $s'_{t+1}$ and store $(s'_t, a_t, r_t, s'_{t+1})$ into $\mathcal{D}'$;
        Randomly sample $(s_{\chi}, a_{\chi}, r_{\chi}, s_{\chi+1})$ from $\mathcal{D}'$;
        Update Critic: $\mathcal{L} = \frac{1}{x}\sum_{x}(\mathcal{Y}_{\chi} - \mathcal{Q}(s_{\chi}, a_{\chi} \mid \theta^{\mathcal{Q}}))^2$;
        Update Actor: $\nabla_{\theta^{\mu}} J \approx \frac{1}{X}\sum_{\chi} \nabla_a \mathcal{Q}(s, a \mid \theta^2)|_{s=s_{\chi}, a=\mu(s_{\chi})} \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu})|_{s=s_{\chi}}$;
        Update target networks: $\theta^{\mathcal{Q}'} \leftarrow v\theta^{\mathcal{Q}} + (1-v)\theta^{\mathcal{Q}'}, \theta^{\mu'} \leftarrow v\theta^{\mu} + (1-v)\theta^{\mu'}$;
    end for
end for
```

4 Experimental Analysis

In this section, we use Chengdu Didi ride-hailing data to evaluate our algorithm against **GREEDY**, **SDE** [6] and **FIX**[1] in terms of platform's profit, the number of served orders and service rate.

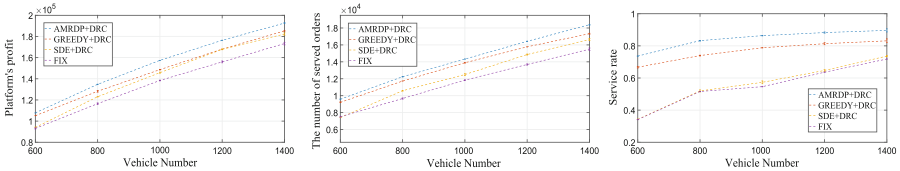
First, we evaluate the effectiveness of **DRC** on the long-term profit by combining it with different pricing algorithms respectively. The results are presented in Fig. 1. It shows that **DRC** can increase the platform's profit by 3.13%, 3.02% and 3.34% respectively when the platform divides the region dynamically, which demonstrates the effectiveness of **DRC** improving the platform's profit.



(a) AMRDP (b) GREEDY (c) SDE

Fig. 1. Impact of dynamic region-clustering on platform's profit

To justify the effectiveness of **AMRDP**, we analyze the results of the four algorithms on the metrics mentioned above, which are presented in Fig. 2. We find that **AMRDP** combined with **DRC** can achieve more profits than all other algorithms from Fig. 2a. Specifically, we can find that the dynamic pricing algorithms(**AMRDP**, **GREEDY** and **SDE**) achieve more profit than **FIX**. It is because **FIX** uses a fixed pricing algorithm that can not adjust prices in response to dynamic changes in supply and demand. From Fig. 2b and 2c, it shows that when the platform combines **AMRDP** with **DRC**, the number of served orders and the service rate are the highest. It indicates that **AMRDP** can serve more orders, which can improve the efficiency of platform services. In summary, the combination of **AMRDP** with **DRC** can achieve more profit, serve more orders, and have a higher service rate with respect to dynamic demand and supply and region-division compared to other algorithms.



(a) Platform's profit (b) Number of served orders (c) Service rate

Fig. 2. Metrics of different algorithms with dynamic region-clustering

5 Conclusion

In this paper, we propose a dynamic region-division based pricing strategy according to demand and supply in different regions, with the goal of maximizing the platform's long-term profit. The experimental results show that the platform's profit is increased when using different pricing algorithms combined with **DRC**, which means that **DRC** can effectively divide regions. Furthermore, we find that the combination of **AMRDP** with **DRC** can bring higher long-term profit, serve more orders and have a higher service rate.

Acknowledgment. This paper was funded by the Humanity and Social Science Youth Research Foundation of Ministry of Education (Grant No. 19YJC790111), the Philosophy and Social Science Post-Foundation of Ministry of Education (Grant No. 18JHQ060) and Shenzhen Fundamental Research Program (Grant No. JCYJ20190809175613332).

References

1. Chen, M., Shen, W., Tang, P., Zuo, S.: Dispatching through pricing: modeling ride-sharing and designing dynamic prices (2019)
2. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)
3. Rempel, J.: A review of uber, the growing alternative to traditional taxi service. AFB AccessWorld® Maga. **51**(6) (2014)
4. Schröder, M., Storch, D.M., Marszal, P., Timme, M.: Anomalous supply shortages from dynamic pricing in on-demand mobility. Nat. Commun. **11**(1), 1–8 (2020)
5. Shi, B., Cao, Z., Luo, Y.: A deep reinforcement learning based dynamic pricing algorithm in ride-hailing. In: Bhattacharya, A., et al. (eds.) DASFAA 2022. LNCS, vol. 13246, pp. 489–505. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-00126-0_36
6. Tong, Y., Wang, L., Zhou, Z., Chen, L., Du, B., Ye, J.: Dynamic pricing in spatial crowdsourcing: a matching-based approach. In: Proceedings of the 2018 International Conference on Management of Data, pp. 773–788 (2018)
7. Yan, C., Zhu, H., Korolko, N., Woodard, D.: Dynamic pricing and matching in ride-hailing platforms. Naval Res. Logist. (NRL) **67**(8), 705–724 (2020)
8. Zhao, Q., Yang, S., Qin, L., Frnti, P.: A grid-growing clustering algorithm for geo-spatial data. Pattern Recogn. Lett. **53**(53), 77–84 (2015)

CANAMRF: An Attention-Based Model for Multimodal Depression Detection

Yuntao Wei, Yuzhe Zhang, Shuyang Zhang, and Hone Zhang(✉)

University of Science and Technology of China, Hefei, China
{yuntaowei,zyz2020,zhangsy2023}@mail.ustc.edu.cn; zhangh@ustc.edu.cn

Abstract. Multimodal depression detection is an important research topic that aims to predict human mental states using multimodal data. Previous methods treat different modalities equally and fuse each modality by naïve mathematical operations without measuring the relative importance between them, which cannot obtain well-performed multimodal representations for downstream depression tasks. In order to tackle the aforementioned concern, we present a **C**ross-modal **A**ttention **N**etwork with **A**daptive **M**ulti-modal **R**ecurrent **F**usion (CANAMRF) for multimodal depression detection. CANAMRF is constructed by a multimodal feature extractor, an Adaptive Multimodal Recurrent Fusion module, and a Hybrid Attention Module. Through experimentation on two benchmark datasets, CANAMRF demonstrates state-of-the-art performance, underscoring the effectiveness of our proposed approach.

Keywords: Depression Detection · Multimodal Representation Learning · Recurrent Fusion

1 Introduction

Depression stands as a prevalent psychiatric disorder while preserving implicit symptoms. Patients haunted by depression often resist timely treatment for fear of misunderstanding from other people, which casts tremendous shade on both their physical and mental health. Recently, a significant amount of research attention has been directed towards the development of multimodal depression assessment systems. These systems leverage diverse cues from text, audio, and video to evaluate depression levels and facilitate diagnostic processes.

However, previous works either focus on single-modality text information or treat each modality equally, and then propose various fusion methods on this basis. Gong et al. [5] utilized topic modeling to partition interviews into segments related to specific topics. Additionally, a feature selection algorithm was employed to retain the most distinctive features. Hanai et al. [1] analyzed data from 142 individuals undergoing depression screening. They employed a Long-Short Term Memory neural network model to detect depression by modeling interactions with audio and text features. Yuan et al. [11] proposed a multimodal

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 111–116, 2024.
https://doi.org/10.1007/978-981-99-7022-3_10

multiorder factor fusion method to exploit high-order interactions between different modalities. This fusion method, which does not discriminate between each modality, cannot well mine the main features that are more effective for depression detection. At the same time, the traditional audio, text, and vision features have not been better in making the category distinction.

In response to these limitations, we introduce a **C**ross-modal **A**ttention **N**etwork with **A**daptive **M**ulti-modal **R**ecurrent **F**usion. CANAMRF first extracts features of four modalities, including textual, acoustic, visual, and the newly proposed sentiment structural modalities, by specific feature extractors separately, then fuses textual features with the other three features through AMRF module. Finally, it utilizes a hybrid attention module to generate distinguishable multimodal representations for subsequent depression detection tasks.

Our primary contributions can be succinctly summarized as follows:

1) We introduce sentiment structural modality as a supplementary modality as a means to augment the performance of multimodal depression detection.
2) We present an innovative approach to modality fusion called Adaptive Multimodal Recurrence Fusion (AMRF). It can dynamically adjust the fusion weights of different modalities, which realizes the trade-off between modalities and has excellent performance.
3) We build a hybrid attention module, which combines cross-modal attention and self-attention mechanisms, to generate representative multimodal features. Extensive experiments and comprehensive analyses are provided to showcase the efficacy of our suggested method.

2 Methodology

In this section, we elucidate the specifics of CANAMRF, illustrated in Fig. 1.

2.1 Feature Extractor

We use specific open-source toolkits and pretrained models, including OpenFace [2], OpenSMILE [4], and BERT [3], for extracting features for textual, visual, and acoustic modalities. In addition, we also introduce a novel high-level semantic structural modality, which consists of five word-level features and three sentence-level features. All features are passed into a 1D temporal convolutional layer to be reshaped into vectors of the same dimention d for subsequent depression detection tasks.

2.2 Adaptive Multi-modal Recurrent Fusion

The detailed framework of Adaptive Multi-modal Recurrent Fusion (AMRF) module is illustrated in Fig. 2(a).

Following Wu et al. [10], we always fuse textual features with features of other modalities (probably visual, acoustic, and sentiment structure), because

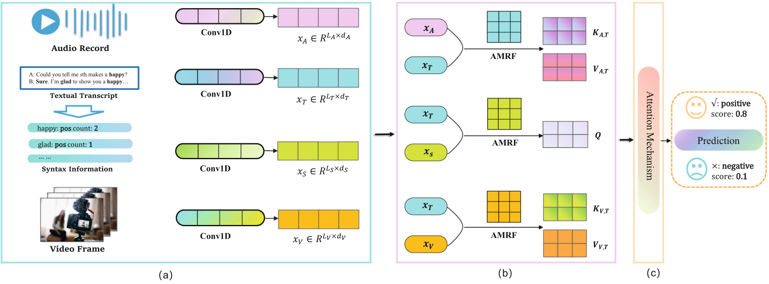


Fig. 1. The overall framework of CANAMRF. (a) Feature extraction procedure for multiple modalities; (b) Fusion of modalities through AMRF module; (c) Hybrid Attention Mechanism.

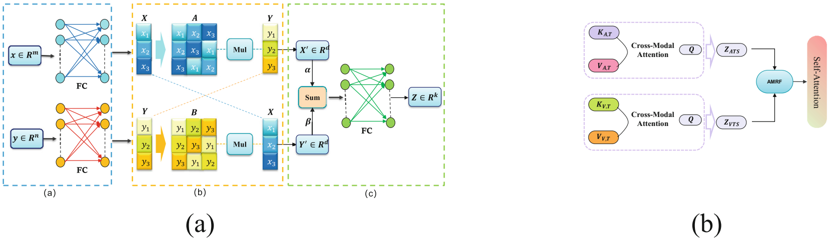


Fig. 2. Subfigure (a): the framework of AMRF module. (a) Features from different modalities are projected into a same low-dimensional space by fully-connected layers; (b) The low-dimensional features are further processed by *Recur* operation; (c) Features are fused according to the adaptive fusion mechanism, and transformed via fully-connected layers to obtain the final representation; Subfigure (b): the framework of Hybrid Attention Module.

of the predominance of textual features. Given two feature vectors of different modalities, for example acoustic characteristics $x \in \mathbb{R}^m$ and textual characteristics $y \in \mathbb{R}^n$, we first map them in a common dimension space d utilizing two projection matrices $W_1 \in \mathbb{R}^{d\times m}$ and $W_2 \in \mathbb{R}^{d\times n} (d \leq \min(m, n))$ by $X = xW_1^\top, \quad Y = yW_2^\top$, where $W_1^\top$ and $W_2^\top$ are the transpose of W_1 and W_2, respectively.

Then we construct recurrent matrices $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times d}$ using projected vectors $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}^d$ by $A = Recur(X), \quad B = Recur(Y)$, where $Recur(\cdot)$ is the operation to construct the recurrent matrix from a vector as visualized in part (b) of Fig. 2(a), which illustrates that all rows of A forms a circular permutation of vector X. In order to fully fuse the elements in the projected vector and recurrent matrix, each row vector of the recurrent matrix is multiplied with the projected vector and then added to the average, as shown in Eq. (1):

$$X^{'} = \frac{1}{d}\sum_{i=1}^{d} a_i \odot A, \quad Y^{'} = \frac{1}{d}\sum_{i=1}^{d} b_i \odot B, \tag{1}$$

where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}^d$ are the ith row vectors of A and B, respectively. $\odot$ denotes the elementwise multiplication.

The final fused features are obtained by $Z = (\alpha X^{'} + \beta Y^{'})W_3^{\top}$, where α and β $(0 \leq \alpha, \beta \leq 1)$ are two learnable weight parameters, and $W_3 \in \mathbb{R}^{d \times k}$ is a projection matrix.

2.3 Hybrid Attention Module

In this subsection, we illustrate the hybrid attention module, whose framework is shown in Fig. 2(b). The hybrid attention module consists of cross-modal attention module, AMRF, and self-attention module.

Following Wu et al. [10], we conduct the attention operation between textual modality and the remaining three modalities. Let $X_{\alpha\beta}$ be the fusion result from modality α and modality β. With the AMRF module, the fusion process can be formulated as:

$$Q = AMRF(X_S, X_T), \tag{2}$$

$$K_{VT} = V_{VT} = X_{VT} = AMRF(X_V, X_T), \tag{3}$$

$$K_{AT} = V_{AT} = X_{AT} = AMRF(X_A, X_T), \tag{4}$$

The cross-modal attention mechanism can be formulated as:

$$Z_{ATS} = CMA(Q, K_{AT}, V_{AT}) = softmax\left(\frac{QK_{AT}^{\top}}{\sqrt{d_k}}\right)V_{AT}, \tag{5}$$

$$Z_{VTS} = CMA(Q, K_{VT}, V_{VT}) = softmax\left(\frac{QK_{VT}^{\top}}{\sqrt{d_k}}\right)V_{VT}, \tag{6}$$

where Z_{ATS} and Z_{VTS} are the output of Cross-Modal Attention. In order to fully integrate four features and have a certain adaptive weight ratio, we pass Z_{ATS} and Z_{VTS} through the AMRF module, followed by a Self-Attention module to get the final fused feature, as shown in Eq. (7):

$$Z_f = Self - Attention(AMRF(Z_{ATS}, Z_{VTS})). \tag{7}$$

2.4 Training Objective

The fused multimodal feature Z_f is flattened and then fed into the fully-connected layers to predict whether a subject has depression or not: $\hat{y} = \sigma(FC(Flatten(Z_f)))$, where $Flatten(\cdot)$ is the flatten operator, $FC(\cdot)$ is the fully-connected layers, and $\sigma(\cdot)$ is an activation function.

We use the Focal loss to train the model: $\mathcal{L}_{fl} = -(1 - \tilde{y})^{\gamma} \log(\tilde{y})$, where $\tilde{y}$ is the estimated probability of being a positive class, and $\gamma \geq 0$ is a tuning parameter.

3 Experiments

In this section, we assess the performance of the proposed CANAMRF using two benchmark datasets, including the Chinese Multimodal Depression Corpus (CMDC) [13] and EATD-Corpus [8] which are frequently used in previous work.

3.1 Baselines

For CMDC and EATD-Corpus, we compare CANAMRF with the following machine learning models: (1) Linear kernel support vector machine (SVM-Linear); (2) SVM based on sequential minimal optimization [7] (SVM-SMO); (3) Logistic regression; (4) Naïve Bayes; (5) Random Forest; (6) Decision Tree; and the following deep learning models: (1) Multimodal LSTM [1]; (2) GRU/BiLSTM-based model [6]; (3) Multimodal Transformer [9]; (4) TAMFN [12].

Table 1. Performance comparison between baseline models and CANAMRF.

Model	CMDC(AT)			CMDC(ATV)			EATD(T)			EATD(A)			EATD(AT)		
	P.	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1	*P.*	*R.*	F_1
SVM-Linear	0.90	0.91	0.91	0.91	0.91	0.91	0.48	**1.00**	0.64	0.54	0.41	0.46	–	–	–
SVM-SMO	0.92	0.91	0.91	0.91	0.89	0.89	–	–	–	–	–	–	–	–	–
Naïve Bayes	0.91	0.89	0.89	0.84	0.84	0.84	–	–	–	–	–	–	–	–	–
Random Forest	–	–	–	–	–	–	0.61	0.53	0.57	0.48	0.53	0.50	–	–	–
Logistic Regression	0.92	0.91	0.91	0.82	0.82	0.82	–	–	–	–	–	–	–	–	–
Decision Tree	–	–	–	–	–	–	0.59	0.43	0.49	0.47	0.44	0.45	–	–	–
Multi-modal LSTM	–	–	–	–	–	–	0.53	0.63	0.57	0.44	0.56	0.49	0.49	0.67	0.57
GRU/BiLSTM-based Model	**0.97**	0.91	0.94	0.87	0.89	0.88	**0.65**	0.66	**0.65**	**0.57**	**0.78**	**0.66**	0.62	0.84	0.71
MulT	0.87	0.96	0.91	**0.97**	0.85	0.91	–	–	–	–	–	–	–	–	–
TAMFN	–	–	–	–	–	–	–	–	–	–	–	–	0.69	**0.85**	0.75
CANAMRF	0.94	**0.97**	**0.95**	0.95	**0.93**	**0.93**	–	–	–	–	–	–	**0.71**	0.83	**0.77**

3.2 Main Results

Table 1 displays the performance comparison of the benchmark models and the CANAMRF model on the CMDC and EATD datasets. For both CMDC and EATD datasets, CANAMRF consistently outperforms the state-of-the-art baselines on F_1 scores, which highlights the effectiveness of CANAMRF, both in unimodal and multimodal depression detection tasks.

4 Conclusion

In this article, we present CANAMRF, a comprehensive framework consisting of three key components. First, we introduce an effective sentiment structural modality as a supplementary modality to enhance the performance of multimodal depression detection tasks. Next, we treat the textual modality as the dominant modality and fuse it with the remaining three modalities using the AMRF module. Finally, we process the fused features using a hybrid attention module to obtain distinct multimodal representations. The experimental results demonstrate the high effectiveness and promising potential of CANAMRF in the detection of depression.

References

1. Al Hanai, T., Ghassemi, M.M., Glass, J.R.: Detecting depression with audio/text sequence modeling of interviews. In: Interspeech, pp. 1716–1720 (2018)
2. Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 59–66. IEEE (2018)
3. 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 (2019)
4. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 1459–1462 (2010)
5. Gong, Y., Poellabauer, C.: Topic modeling based multi-modal depression detection. In: Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge, pp. 69–76 (2017)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput. **13**(3), 637–649 (2001)
8. Shen, Y., Yang, H., Lin, L.: Automatic depression detection: an emotional audio-textual corpus and a gru/bilstm-based model. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6247–6251. IEEE (2022)
9. Tsai, Y.H.H., Bai, S., Liang, P.P., Kolter, J.Z., Morency, L.P., Salakhutdinov, R.: Multimodal transformer for unaligned multimodal language sequences. In: Proceedings of the conference. Association for Computational Linguistics, Meeting, vol. 2019, p. 6558. NIH Public Access (2019)
10. Wu, Y., Lin, Z., Zhao, Y., Qin, B., Zhu, L.N.: A text-centered shared-private framework via cross-modal prediction for multimodal sentiment analysis. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 4730–4738 (2021)
11. Yuan, C., Xu, Q., Luo, Y.: Depression diagnosis and analysis via multimodal multi-order factor fusion. arXiv preprint arXiv:2301.00254 (2022)
12. Zhou, L., Liu, Z., Shangguan, Z., Yuan, X., Li, Y., Hu, B.: TAMFN: time-aware attention multimodal fusion network for depression detection. IEEE Trans. Neural Syst. Rehabil. Eng. **31**, 669–679 (2022)
13. Zou, B., et al.: Semi-structural interview-based Chinese multimodal depression corpus towards automatic preliminary screening of depressive disorders. IEEE Trans. Affect. Comput. (2022)

CASSOR: Class-Aware Sample Selection for Ordinal Regression with Noisy Labels

Yue Yuan[1,2,3], Sheng Wan[1,2,3], Chuang Zhang[1,2,3], and Chen Gong[1,2,3](✉)

[1] School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
[2] Key Laboratory of Intelligent Perception and Systems
for High-Dimensional Information of Ministry of Education, Nanjing, China
[3] Jiangsu Key Laboratory of Image and Video Understanding for Social Security,
Nanjing, China
chen.gong@njust.edu.cn

Abstract. Ordinal regression aims at solving the classification problem, where the categories are related in a natural order. Due to the difficulty in distinguishing between highly relevant categories, label noise is frequently present in ordinal data. Moreover, the varying degrees of relevance between categories can lead to an inconsistent distribution of misclassification loss across categories, posing a challenge to select clean data consistently from all categories for training. To overcome this limitation, we develop a sample selection method termed '**C**lass-**A**ware **S**ample **S**election for **O**rdinal **R**egression' (CASSOR). To be concrete, we devise a class-specific sample selection strategy in order to adaptively acquire sufficient clean examples for robust model training. Moreover, a label-ranking regularizer is designed to help guide the sample selection process via exploring the ordinal relationship between different examples. As a result, our proposed CASSOR is endowed with strong discrimination abilities on ordinal data. Intensive experiments have been performed on multiple real-world ordinal regression datasets, which firmly demonstrates the effectiveness of our method.

Keywords: Ordinal regression · Label noise · Weakly-supervised learning

1 Introduction

Ordinal regression, also known as ordinal classification, aims to predict categories on an ordinal scale [5]. Unlike the nominal classification setting, ordinal regression involves labels that naturally possess a specific order [6]. To now, ordinal regression has found its applications in various fields, such as age estimation [2]. The existing methods to deal with ordinal regression tasks can be roughly divided into two types, namely regression and classification. The regression approaches aim to predict the values of the latent variable by mapping the input space to a one-dimensional real space [3] before predicting the categories of the input examples. The classification approaches, on the other hand, embed the ordinal relationship between categories into loss functions [12], labels [4, 17], or architectural design [16].

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 117–123, 2024.
https://doi.org/10.1007/978-981-99-7022-3_11

The existing ordinal regression techniques are primarily designed for clean-label settings. However, the class labels observed in ordinal data may not always be correct. This is because the potential relevance between adjacent categories will make it challenging for annotators to accurately distinguish between different categories. As a result, the label noise can probably lead to performance degradation in model training. To now, various deep learning approaches have been proposed for handling classification problems with label noise. Most of them focus on the estimation of the noise transition matrix [15] or the selection of clean examples [8,14]. The former aims to employ the transition matrix to build a risk-consistent estimator or a classifier-consistent estimator, while obtaining an accurate noise transition matrix can be challenging in practical scenarios [7]. Here, [5] is the only method designed for ordinal regression under label noise, which uses the noise transition matrix to construct the unbiased estimator of the true risk. On the other hand, the sample selection methods focus on selecting clean examples for model training and yield relatively satisfactory performance [8]. They usually predefine a loss threshold heuristically to regulate the number of clean examples, assuming that examples with loss below the threshold are probably clean [8,10].

Nevertheless, the above-mentioned sample selection methods are designed for nominal classification problems, which fail to exploit the fundamental characteristics of ordinal data. To be specific, if a category is highly relevant to its neighbors, it can be misclassified with a high probability, which leads to a large misclassification loss. Meanwhile, if a category is weakly related to its neighbors, the corresponding misclassification loss could be small. This will result in inconsistent distribution of misclassification loss across categories. Simply selecting the small-loss examples with a single threshold can lead to imbalanced sample selection across categories. As a result, highly relevant categories cannot provide sufficient information for model learning, ultimately degrading the model performance. In addition, ordinal data typically exhibit a natural label order that benefits the learning of ordinal models [6], which is, however, neglected by the nominal classification methods.

In light of the aforementioned challenges, we propose a new type of sample selection method termed **C**lass-**A**ware **S**ample **S**election for **O**rdinal **R**egression (CASSOR). Firstly, we design a class-aware sample selection strategy via calculating a class-specific score for each category. The score determines the number of examples chosen from each category, ensuring that categories with significant misclassification contribute adequate examples for model training. Considering the varying misclassification loss associated with different categories during the training phase, the class-specific score can be dynamically adjusted. This could also help prevent the model from overfitting to certain noisy examples and improve the generalization abilities. Additionally, since a biased selection of training examples is inevitable [7], we employ a dual-network architecture. As such, the potential errors caused by the biased selection can be reduced by the dual networks in a mutually beneficial manner [8]. Furthermore, to incorporate the inherent ordinal relationship between labels, we design a new type of OT loss called 'Optimal Transport regularized by label Ranking' (OTR). Unlike the traditional OT loss [1,12,16], which neglects the ordinal relationship among examples, our proposed OTR loss preserves the label order between the predicted results of the dual networks.

Therefore, the inherent ordinal relationship can help guide the sample selection process and further reduce the accumulated errors caused by the biased sample selection.

2 Our Method

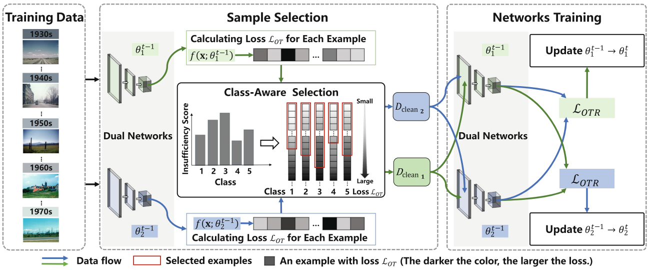

Training Data
1930s
1940s
1950s
1960s
1970s
Sample Selection
Dual Networks
θ_1^{t-1}
θ_2^{t-1}
Calculating Loss $\mathcal{L}_{OT}$ for Each Example
$f(\mathbf{x};\theta_1^{t-1})$
Class-Aware Selection
Insufficiency Score
1 2 3 4 5
Class
Small
Large
Class 1 2 3 4 5 Loss $\mathcal{L}_{OT}$
$f(\mathbf{x};\theta_2^{t-1})$
Calculating Loss $\mathcal{L}_{OT}$ for Each Example
$D_{\text{clean }2}$
$D_{\text{clean }1}$
Networks Training
Dual Networks
θ_1^{t-1}
θ_2^{t-1}
Update $\theta_1^{t-1} \rightarrow \theta_1^t$
$\mathcal{L}_{OTR}$
$\mathcal{L}_{OTR}$
Update $\theta_2^{t-1} \rightarrow \theta_2^t$
Data flow
Selected examples
An example with loss $\mathcal{L}_{OT}$ (The darker the color, the larger the loss.)

Fig. 1. The pipeline of our proposed method.

2.1 Preliminaries

In ordinal regression problems, the label of an example with a feature vector $\mathbf{x}$ is denoted as y, where $y \in \mathcal{Y} = \{1, 2, \ldots, K\}$. That is, y is in a label space with K different labels, and the class labels satisfy $1 \prec 2 \prec \ldots \prec K$ with '$\prec$' representing order relation. The objective of ordinal regression is to find a classification rule or function to predict the categories of new examples given a training set of N examples, namely $D = \{\mathbf{x}_i, y_i\}_{i=1}^N$. Label noise refers to the situation that the observed label does not match the ground-truth label y^*, *i.e.*, $y \neq y^*$. To ensure the unimodality of model prediction, we adopt the architecture in the ordinal regression model [16] as the backbone of our method and the baseline methods. Let $f(\cdot;\theta)$ be the latent function for the network parameterized by θ. Furthermore, the Optimal Transport (OT) loss [12,16] of the example $\mathbf{x}_i$ is employed to measure the misclassification in ordinal regression tasks:

$$\mathcal{L}_{OT}(f(\mathbf{x}_i;\theta), y_i) = \sum_{k=1}^{K} d(y_i, k)\, f_k(\mathbf{x}_i;\theta), \tag{1}$$

wherein $d(y_i, k) = |y_i - k|^m$ measures the label distance between y_i and k with $m \geq 1$.

2.2 Overall Framework

As shown in Fig. 1, the proposed method consists of two critical components which are designed for ordinal regression with label noise: (1) Class-Aware selection strategy, which adaptively selects reliable data from each category for robust modeling training (see Sect. 2.3); (2) Regularization with label ranking, which aims to incorporate the label order inherently contained in ordinal data for model learning (see Sect. 2.4).

2.3 Class-Aware Sample Selection

We develop a Class-Aware sample selection strategy for ordinal data in order to sufficiently learn from the categories with much misclassification. Firstly, we aim to compute an insufficiency score which can be obtained based on the distance between the average distribution of the prediction from each category and the distribution of each target category. Here, the average distribution $p_{y=k}$ of each observed category can be calculated by $p_{y=k} = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{1}[y_i = k] f\left(\mathbf{x}_i; \theta\right)$, where $f\left(x_i; \theta\right)$ indicates the predicted probability distribution of the example $\mathbf{x}_i$ by network parameters θ, and N_k is the number of examples in the k-th category. The j-th element in $p_{y=k}$ represents the probability of predicting an example of the k-th category to the j-th category. After that, we use Jensen-Shannon Divergence (JSD) [11] represented as $JS(\cdot||\cdot)$ to measure the dissimilarity between the average distribution and Dirac point mass [16] characterized by a one-hot probability mass function. We choose JSD because it is relatively simple and efficient for computation. A smaller JSD value indicates that the two distributions are more similar to each other. On this basis, we construct a matrix $\mathbf{S}$ with $\mathbf{S}_{i,j}$ denoting the JSD between the average distribution of prediction related to the i-th category and the one-hot distribution $Dirac\left(j\right)$ of class j:

$$\mathbf{S}_{i,j} = JS\left(p_{y=i}||Dirac\left(j\right)\right), \quad \forall i, j \in \{1, ..., K\}. \tag{2}$$

With Eq. (2), we can obtain the insufficiency score for the j-th category, which is expressed as $v_j = \frac{1}{K} \sum_{i=1}^{K} \mathbf{S}_{i,j}$. Here, the insufficiency score can be used to measure misclassification in a specific category, and a larger score often corresponds to more misclassified examples. For practical use, the insufficiency score is normalized as follows in order to eliminate the influences of different scales: $\tilde{v}_j = \frac{v_j - mean(v)}{max(v) - min(v)}$, where $\tilde{v}_j \in (-1, 1)$. The normalized insufficiency score can then be utilized to adjust the ratio of selected examples for each category. Concretely, the selection ratio of the j-th category is presented as $\mathbf{r}_j = 0.5 + \lambda \times \tilde{v}_j$, where λ is a hyperparameter assigned to the insufficiency score $\tilde{v}_j$. Afterward, we select the examples with small classification loss, *i.e.*, D_{clean}, based on the ratio $\mathbf{r}_j$ at each epoch, so that the model can be encouraged to learn from the categories with relatively much misclassification. Note that the misclassification loss is measured by OT loss [12, 16] in our method. Consequently, the model's discrimination abilities towards prone-to-misclassification categories will be enhanced.

2.4 Regularization with Label Ranking

We believe the ordering information of ordinal labels can enhance the performance of the model in ordinal regression tasks [6]. To this end, we have introduced an OTR loss that aims to maintain the label ranking between the predicted results of the dual networks, which consists of the traditional OT loss and a label-ranking loss $\mathcal{L}_{LR}$. Different from OT loss which focuses on the individual example, the proposed OTR loss concentrates on the relationship between each pair of examples. Here, the OTR loss is:

$$\mathcal{L}_{OTR} = \tilde{\mathcal{L}}_{OT} + \beta \times \mathcal{L}_{LR}, \tag{3}$$

where β is a hyperparameter and $\tilde{\mathcal{L}}_{OT}$ represents the average OT loss of the selected examples. The label-ranking loss $\mathcal{L}_{LR}$ in Eq. (3) can be expressed as

$$\mathcal{L}_{LR} = \sum_{k=1}^{K-1} \frac{\sum_{d(y_i,y_j)=k} JS(f(\mathbf{x}_i;\theta_1), f(\mathbf{x}_j;\theta_2))}{\sum_{d(y_i,y_j)\geq k} JS(f(\mathbf{x}_i;\theta_1), f(\mathbf{x}_j;\theta_2))}, \tag{4}$$

where $f\left(\mathbf{x}_i;\theta_1\right)$ is the prediction of $\mathbf{x}_i$ generated from the network parameterized by θ_1, and so on. In Eq. (4), $d\left(\cdot,\cdot\right)$ is the label distance function also used in Eq. (1). The objective of the Eq. (4) is to enforce a condition where given a pair of examples, the estimated distribution distance between the sample pair with a greater label distance is larger than the estimated distribution distance between the sample pair with a smaller label distance. Finally, the OTR loss of Eq. (3) is used to update the network parameters.

3 Experiments

3.1 Experimental Settings

Datasets. Given the research emphasis on ordinal regression and label noise, we adhere to established practices [4,16] by using three standard datasets for assessment: Historical Color Image (HCI), Adience, and Diabetic Retinopathy (DR). HCI [13] comprises 1,325 images for a five-class ordinal task spanning the '1930s' to '1970s'. Adience [9] focuses on age estimation, where we selected and adapted the first six age groups from train-test splits[1]. DR[2] includes retinal images with diabetic retinopathy categorized into severity levels. We categorize and adapt it into 'no DR,' 'Mild,' 'Moderate,' and 'Severe DR and Proliferative DR,' and 1,680 images per class are used for evaluation.

Ordinal Label Noise Generation. Similar to [5], we hold the assumption that the probability of mislabeling decreases with the increase of label distance. By letting $\mathbf{T}$ be the noise transition matrix and ρ be the total noise rate, $\mathbf{T}_{i,j}$ denotes the probability of flipping label i to label j, $\mathbf{T}_{i,i} = 1 - \rho(i \in \{1,\ldots,K\})$, and $\mathbf{T}_{i,j}(i \neq j)$ can be calculated as $\mathbf{T}_{i,j} = \rho\frac{e^{\mathbf{T}^*_{i,j}}}{\sum_{k=1}^{K} e^{\mathbf{T}^*_{i,k}}}$. Here, $\mathbf{T}^*_{i,j}$ follows the standard normal distribution and can be formulated as $\mathbf{T}^*_{i,j} = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{1}{2}}$, where σ is set to 2 empirically.

Baseline Methods. We evaluate the effectiveness of our method by comparing it with multiple representative approaches, including the typical ordinal methods such as UNIORD [16], SORD [4], and CORAL [2]; label noise-robust learning algorithms such as forward correction (F-correction), backward correction (B-correction) [15], and Co-teaching [8]; and label noise-robust ordinal regression methods such as RDORLN [5]. For RDORLN [5], we use the same ordinal regression loss as our method.

3.2 Experimental Results

For HCI, we evaluate the proposed method with synthetic label noise, where the noise rate ρ is chosen from $\{0.2, 0.4\}$. We run five individual trials for all compared methods under each noise level and report the mean MAE, RMSE, and standard deviation

[1] https://github.com/GilLevi/AgeGenderDeepLearning/tree/master/Folds.

[2] https://www.kaggle.com/c/diabetic-retinopathy-detection/data.

Table 1. Experimental results of all compared methods on noisy HCI, Adience, and DR.

Dataset	ρ		UNIORD [16]	SORD [4]	CORAL [2]	F-correction [15]	B-correction [15]	Co-teaching [8]	RDORLN [5]	**Our method**
HCI	0.2	MAE↓	0.790 ± 0.039	0.767 ± 0.047	0.935 ± 0.058	0.782 ± 0.037	0.862 ± 0.026	0.809 ± 0.049	0.777 ± 0.025	**0.642 ± 0.029**
		RMSE↓	1.224 ± 0.038	1.200 ± 0.058	1.377 ± 0.069	1.196 ± 0.048	1.301 ± 0.058	1.261 ± 0.061	1.221 ± 0.041	**1.047 ± 0.044**
	0.4	MAE↓	0.990 ± 0.044	0.998 ± 0.069	1.100 ± 0.082	0.972 ± 0.031	1.106 ± 0.084	1.020 ± 0.075	1.032 ± 0.049	**0.728 ± 0.074**
		RMSE↓	1.403 ± 0.057	1.418 ± 0.074	1.555 ± 0.090	1.382 ± 0.015	1.573 ± 0.103	1.487 ± 0.088	1.467 ± 0.071	**1.137 ± 0.070**
Adience	0.2	MAE↓	0.566 ± 0.043	0.566 ± 0.032	0.810 ± 0.081	0.533 ± 0.030	0.533 ± 0.043	0.423 ± 0.040	0.543 ± 0.039	**0.407 ± 0.030**
		RMSE↓	0.898 ± 0.046	0.886 ± 0.033	1.125 ± 0.075	0.876 ± 0.037	0.861 ± 0.048	0.737 ± 0.043	0.881 ± 0.039	**0.704 ± 0.038**
	0.4	MAE↓	0.759 ± 0.037	0.797 ± 0.053	0.947 ± 0.072	0.812 ± 0.060	0.811 ± 0.070	0.492 ± 0.033	0.786 ± 0.022	**0.420 ± 0.035**
		RMSE↓	1.091 ± 0.045	1.150 ± 0.057	1.301 ± 0.065	1.264 ± 0.086	1.180 ± 0.074	0.797 ± 0.028	1.149 ± 0.031	**0.715 ± 0.037**
DR	0.2	MAE↓	0.636 ± 0.015	0.651 ± 0.021	0.730 ± 0.011	0.635 ± 0.017	0.673 ± 0.016	0.597 ± 0.025	0.653 ± 0.011	**0.577 ± 0.016**
		RMSE↓	0.911 ± 0.014	0.948 ± 0.017	1.041 ± 0.015	0.917 ± 0.019	0.973 ± 0.019	0.927 ± 0.030	0.936 ± 0.012	**0.862 ± 0.020**
	0.4	MAE↓	0.769 ± 0.012	0.775 ± 0.017	0.848 ± 0.017	0.777 ± 0.016	0.792 ± 0.021	0.617 ± 0.020	0.762 ± 0.019	**0.609 ± 0.012**
		RMSE↓	1.029 ± 0.011	1.057 ± 0.025	1.155 ± 0.024	1.048 ± 0.020	1.093 ± 0.029	0.943 ± 0.030	1.029 ± 0.013	**0.902 ± 0.021**

in Table 1. Note that the performance of the ordinal regression methods consistently decreases as the noise level increases. Particularly, RDORLN [5] achieves poor results due to its reliance on the assumption that the noise transition matrix accurately reflects the true-noisy label relationship, which may not hold for the HCI dataset. In contrast, our method consistently achieves good results, showcasing its effectiveness across all noise rates. For Adience, the ordinal regression methods, such as UNIORD, SORD, and CORAL, exhibit unsatisfactory performance as a result of their inability to address label noise. Similarly, the label noise-robust methods, such as B-correction and Co-teaching, also yield poor results due to the inadequate consideration of ordinal information. We also performed well on DR, especially in RMSE.

Table 2. Experimental results of the proposed method with different key components.

Dataset	ρ	MAE↓				RMSE↓			
		A	B	C	D	A	B	C	D
DR	0.2	0.632 ± 0.020	0.593 ± 0.008	0.579 ± 0.011	0.577 ± 0.016	0.964 ± 0.015	0.888 ± 0.015	0.879 ± 0.017	0.862 ± 0.020
	0.4	0.671 ± 0.006	0.632 ± 0.010	0.621 ± 0.013	0.609 ± 0.012	0.983 ± 0.007	0.922 ± 0.008	0.908 ± 0.006	0.902 ± 0.021

3.3 Ablation Study

Our method includes three crucial elements, namely the class-aware sample selection, the dual-network architecture, and the label-ranking regularization. We incrementally add these key components from **A** to **D**. **A**: A naïve baseline method, where 50% of small-loss examples over all the training data are selected for training. **B**: Incorporating the class-aware selection strategy. **C**: Incorporating the dual-network architecture. **D**: Incorporating the regularization with label ranking equipped with dual-network. The experimental results are shown in Table 2. As expected, the performance of the model can be improved when each component is applied.

4 Conclusion

In this paper, we introduce CASSOR, a novel sample selection approach for handling label noise in ordinal regression. CASSOR aims to mitigate the negative effects of inconsistent misclassification loss in the sample selection of ordinal data. Furthermore, a label-ranking regularizer is devised to guide the sample selection process with ordinal relations. As a result, our proposed method demonstrates strong performance on various real-world ordinal datasets. Future work will focus on developing a robust quantitative framework for measuring the essential differences between ordinal class labels.

Acknowledgement. This research is supported by NSF of Jiangsu Province (Nos: BZ2021013, BK20220080).

References

1. Beckham, C., Pal, C.: Unimodal probability distributions for deep ordinal classification. In: ICML, pp. 411–419 (2017)
2. Cao, W., Mirjalili, V., Raschka, S.: Rank consistent ordinal regression for neural networks with application to age estimation. Pattern Recogn. Lett. **140**, 325–331 (2020)
3. Chu, W., Ghahramani, Z., Williams, C.K.: Gaussian processes for ordinal regression. J. Mach. Learn. Res. **6**(7), 1019–1041 (2005)
4. Diaz, R., Marathe, A.: Soft labels for ordinal regression. In: CVPR, pp. 4738–4747 (2019)
5. Garg, B., Manwani, N.: Robust deep ordinal regression under label noise. In: ACML, pp. 782–796 (2020)
6. Gutiérrez, P.A., Pérez-Ortiz, M., Sánchez-Monedero, J., Fernández-Navarro, F., Hervás-Martinez, C.: Ordinal regression methods: survey and experimental study. TKDE **28**(1), 127–146 (2015)
7. Han, B., et al.: A survey of label-noise representation learning: past, present and future. arXiv:2011.04406 (2020)
8. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: NeurIPS, vol. 31 (2018)
9. Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. In: CVPRW, pp. 34–42 (2015)
10. Li, J., Socher, R., Hoi, S.C.: DivideMix: learning with noisy labels as semi-supervised learning. In: ICLR (2019)
11. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theor. **37**(1), 145–151 (1991)
12. Liu, X., Han, X., Qiao, Y., Ge, Y., Li, S., Lu, J.: Unimodal-uniform constrained Wasserstein training for medical diagnosis. In: ICCVW (2019)
13. Palermo, F., Hays, J., Efros, A.A.: Dating historical color images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 499–512. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_36
14. Patel, D., Sastry, P.: Adaptive sample selection for robust learning under label noise. In: WACV, pp. 3932–3942 (2023)
15. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: CVPR, pp. 1944–1952 (2017)
16. Shaham, U., Svirsky, J.: Deep ordinal regression using optimal transport loss and unimodal output probabilities. arXiv:2011.07607 (2020)
17. Vargas, V.M., Gutiérrez, P.A., Barbero-Gómez, J., Hervás-Martínez, C.: Soft labelling based on triangular distributions for ordinal classification. Inf. Fus. **93**, 258–267 (2023)

Incomplete Multi-view Weak-Label Learning with Noisy Features and Imbalanced Labels

Zhiwei Li[1], Zijian Yang[1](✉), Lu Sun[1], Mineichi Kudo[2], and Keigo Kimura[2]

[1] ShanghaiTech University, 393 Middle Huaxia Road, Pudong, Shanghai, China
{lizhw,yangzj,sunlu1}@shanghaitech.edu.cn
[2] Hokkaido University, Kita 8, Nishi 5, Kita-ku, Sapporo, Hokkaido, Japan
{mine,kimura5}@ist.hokudai.ac.jp

Abstract. A variety of modern applications exhibit multi-view multi-label learning, where each sample has multi-view features, and multiple labels are correlated via common views. Current methods usually fail to directly deal with the setting where only a subset of features and labels are observed for each sample, and ignore the presence of noisy views and imbalanced labels in real-world problems. In this paper, we propose a novel method to overcome the limitations. It jointly embeds incomplete views and weak labels into a low-dimensional subspace with adaptive weights, and facilitates the difference between embedding weight matrices via auto-weighted Hilbert-Schmidt Independence Criterion (HSIC) to reduce the redundancy. Moreover, it adaptively learns view-wise importance for embedding to detect noisy views, and mitigates the label imbalance problem by focal loss. Experimental results on four real-world multi-view multi-label datasets demonstrate the effectiveness of the proposed method.

Keywords: Multi-View Multi-Label Learning · Weakly Supervised Learning · Hilbert-Schmidt Independence Criterion · Focal Loss

1 Introduction

In many real-world applications, samples are often represented by several feature subsets, and meanwhile associated with multiple labels [10]. In addition, it is probably that only a subset of features and labels are observed for each sample. Current related methods [5,12] usually treat multiple view equally and complete the missing data by encouraging low-rankness, which may not hold in practice.

To address the challenge, we propose a novel method for i**N**complete multi-view we**A**k-label learning with no**I**sy features and imba**L**anced labels (**NAIL**). NAIL tackles the problem by projecting multiple incomplete views into a common latent subspace using the $L_{2,1}$ norm, adaptively adjusting view-wise weights to detect noisy views. It also embeds weak labels into the same subspace, employing Focal Loss to handle label imbalance. To remove the redundancy during the embeding, NAIL utilizes the auto-weighted Hilbert-Schmidt Independence Criterion (HSIC) to drive embedding weight matrices to differ from each other in Reproducing Kernel Hilbert Spaces (RKHSs). The workflow of NAIL is illustrated in Fig. 1.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 124–130, 2024.
https://doi.org/10.1007/978-981-99-7022-3_12

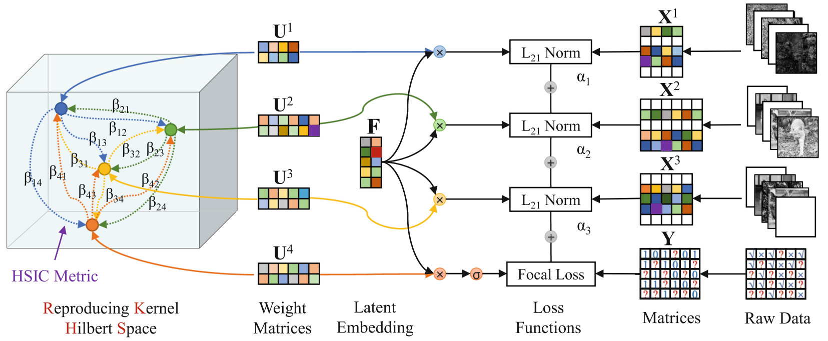


Fig. 1. The framework of NAIL. NAIL first reconstructs incomplete views $\{\mathbf{X}^v\}_{v=1}^m$ and weak labels $\mathbf{Y}$ by a common low-dimensional representation $\mathbf{F}$, i.e., $\mathbf{X}^v \approx \mathbf{F}\mathbf{U}^v(\forall v)$ and $\mathbf{Y} \approx \sigma(\mathbf{F}\mathbf{U}^{m+1})$, where σ denotes the sigmoid function. The reconstruction errors for $\{\mathbf{X}^v\}_{v=1}^m$ and $\mathbf{Y}$ are measured by $L_{2,1}$-norm and focal loss, respectively, and are adaptively weighted by $\{\alpha_v\}_{v=1}^m$. It then projects weight matrices $\{\mathbf{U}^v\}_{v=1}^{m+1}$ into RKHSs and promotes the differences between weight matrices via $\boldsymbol{\beta}$ auto-weighted HSIC, in order to reduce the redundancy during embedding. Finally, NAIL predicts unobserved labels in $\mathbf{Y}$ based on $\sigma(\mathbf{F}\mathbf{U}^{m+1})$.

2 Methodology

Let $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \ldots, \mathbf{x}_n^v]^T \in \mathbb{R}^{n\times d_v}$ denote the feature matrix in the v-th view, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^T \in \{0,1\}^{n\times l}$ denote the label matrix, where $y_{ij} = 1$ means that the j-th label is assigned to the i-th instance and $y_{ij} = 0$ otherwise. We introduce $\mathbf{O}_\mathbf{X}^v \in \mathbb{R}^{n\times d_v}$ and $\mathbf{O}_\mathbf{Y} \in \mathbb{R}^{n\times l}$ to denote indices of the entries in $\mathbf{X}^v$ and $\mathbf{Y}$, respectively, such that $(\mathbf{O}_\mathbf{X}^v)_{ij} = 1$ or $(\mathbf{O}_\mathbf{Y})_{ij} = 1$ if the (i,j)-th entry is observed in $\mathbf{X}^v$ or $\mathbf{Y}$, and $(\mathbf{O}_\mathbf{X}^v)_{ij} = 0$ or $(\mathbf{O}_\mathbf{Y})_{ij} = 0$ otherwise. The goal of NAIL is to predict unobserved labels in presence of both noisy views and imbalanced labels.

2.1 Auto-weighted Incomplete Multi-view Embedding

Given a multi-view dataset, we seek to find a shared latent subspace $\mathbf{F} \in \mathbb{R}^{n\times k}$ ($k < d_v$, $\forall v$) by integrating complementary information from different views [3], which can be formulated as $\min_{\mathbf{F},\{\mathbf{U}^v\}\geq 0} \sum_{v=1}^m ||\mathbf{X}^v - \mathbf{F}\mathbf{U}^v||_F^2$, where $||\cdot||_F$ represents the Frobenius norm and $\mathbf{U}^v \in \mathbb{R}^{k\times d_v}$ is the weight matrix of the v-th view. It embeds multiple views into an identical subspace by treating each view equally, deviating from the true latent subspace when multiple views have different importance during embedding. Furthermore, the existence of missing entries poses another challenge. To address the problems, we propose the auto-weighted incomplete multi-view embedding:

$$\min_{\substack{\boldsymbol{\alpha},\mathbf{F},\{\mathbf{U}^v\}\geq 0,\\ \sum \alpha_v = 1}} \sum_{v=1}^{m} \alpha_v^s ||\mathbf{O}_\mathbf{X}^v \odot (\mathbf{X}^v - \mathbf{F}\mathbf{U}^v)||_{2,1} \tag{1}$$

where $\odot$ is the Hadamard product, and $||\mathbf{A}||_{2,1} = \sum_{i=1}^n ||\mathbf{a}_{i:}||_2$ represents the $L_{2,1}$ norm, which is insensitive to outlier samples by decreasing the contribution of the outlier to the reconstruction error. In (1), α_v is introduced to weight the embedding importance of the v-th view ($\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_m]$), and s is a constant, which is fixed as 0.5 in experiments. According to (1), $\mathbf{X}^v$ is mapped to a common latent representation $\mathbf{F} \in \mathbb{R}^{n\times k}$ with view-specific adaptive weight α_v. For the v-th view, the more importance contributed to embedding $\mathbf{F}$, the higher weight of α_v, and vice versa.

2.2 Imbalanced Weak-Label Embedding

Cross Entropy (CE) [2] is often used to measure the classification loss between the ground truth and predictions. However, possible label imbalance, i.e., a large difference between the proportions of positive and negative labels, can lead to a drop in prediction accuracy. Here we adopt Focal Loss (FL) [6] to mitigate this problem. For the j-th label in the i-th sample, focal loss $\text{FL}(y_{ij}, p_{ij})$ is computed based on the ground truth y_{ij} and the predicted label probability p_{ij}, i.e., $\text{FL}(y_{ij}, p_{ij}) = -a_{ij}(1 - q_{ij})^{\gamma} \log(q_{ij})$, where γ is a constant, and a_{ij} takes a value $a \in [0, 1]$ if $y_{ij} = 1$ and $a_{ij} = 1 - a$ otherwise. In experiments, we fix $\gamma = 2$ and $a = 0.5$. In focal loss, $q_{ij} = p_{ij}$ if $y_{ij} = 1$, and $q_{ij} = 1 - p_{ij}$ otherwise. Predicted probability p_{ij} is calculated by $p_{ij} = \sigma(\mathbf{f}_{i:}^T \mathbf{u}_{:j}^{m+1})$, where $\sigma(\cdot)$ is the sigmoid function, $\mathbf{f}_{i:}$ is the i-th row of the latent embedding $\mathbf{F}$ in (1), and $\mathbf{u}_{:j}^{m+1}$ is the j-th column of the weight matrix $\mathbf{U}^{m+1}$ for label embedding. Therefore, imbalanced weak-label embedding can be modeled as follows:

$$\min_{\mathbf{F},\mathbf{U}^{m+1}\geq 0} \sum_{(i,j)\in \mathbf{O}_Y} \text{FL}(y_{ij}, \sigma(\mathbf{f}_{i:}^T \mathbf{u}_{:j}^{m+1})). \tag{2}$$

Thus, the label imbalance problem is alleviated by applying focal loss on the observed labels, which helps the model to focus on learning hard misclassified samples.

2.3 Correlation Modeling by Auto-weighted HSIC

Next, we adopt the Hilbert-Schmidt Independence Criterion (HSIC) [4] to model the nonlinear correlations among weight matrices $\{\mathbf{U}^v\}_{v=1}^{m+1}$ in an adaptive manner. Specifically, HSIC estimates the dependency between $\mathbf{U}^v$ and $\mathbf{U}^{v'}$ ($v' \neq v$) in the Reproducing Kernel Hilbert Spaces (RKHSs), i.e., $\text{HSIC}(\mathbf{U}^{\text{v}}, \mathbf{U}^{\text{v}'}) = (\text{n}-1)^{-2}\text{tr}(\mathbf{K}^{\text{v}}\mathbf{H}\mathbf{K}^{\text{v}'}\mathbf{H})$, where $\mathbf{K}^v \in \mathbb{R}^{n\times n}$ is the Gram matrix that measures the similarity between row vectors of $\mathbf{U}^v$. $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbb{1}\mathbb{1}^T$ is the centering matrix, where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is an identity matrix, and $\mathbb{1} \in \mathbb{R}^n$ is an all-one vector. It is guaranteed that the lower the value of HSIC, the lower the dependence between $\mathbf{U}^v$ and $\mathbf{U}^{v'}$. Thus, to reduce the redundancy among $\mathbf{U}^v$s during embedding, we can minimize the HSIC between each pair of weight matrices. However, noisy views make directly minimizing the HSIC too restrictive in practice. To address the problem, we propose to minimize auto-weighted HSIC, i.e.,

$$\min_{\substack{\boldsymbol{\beta},\{\mathbf{U}^v\}\geq 0\\ ||\boldsymbol{\beta}_v||_2=1}} \sum_{v=1}^{m+1} \sum_{v'\neq v} \beta_{vv'} \text{HSIC}(\mathbf{U}^{\text{v}}, \mathbf{U}^{\text{v}'}) \tag{3}$$

where $\beta_{vv'} \geq 0$ measures the importance of the correlation between $\mathbf{U}^v$ and $\mathbf{U}^{v'}$ and $\boldsymbol{\beta}_v = [\beta_{v1}, \beta_{v2}, \ldots, \beta_{v(m+1)}]$. Once the v-th view is indeed noisy, a relatively larger value will be assigned to β_v, leading to the decorrelation between $\mathbf{U}^v$ and $\mathbf{U}^{v'}$ $(\forall v' \neq v)$ by imposing a stronger degree of penalty on HSIC. Therefore, multiple views and labels are correlated in a non-linear and adaptive way.

2.4 The Proposed NAIL Method

By incorporating (1), (2) and (3), we now have the optimization problem of NAIL:

$$\min_{\substack{\boldsymbol{\alpha},\boldsymbol{\beta},\\ \mathbf{F},\{\mathbf{U}^v\}}} \sum_{v=1}^{m} \alpha_v^s ||\mathbf{O}_{\mathbf{X}}^v \odot (\mathbf{X}^v - \mathbf{F}\mathbf{U}^v)||_{2,1} + \lambda \sum_{(i,j)\in\mathbf{O}_Y} \mathrm{FL}(y_{ij}, \sigma(\mathbf{f}_{i:}^T \mathbf{u}_{:j})) \tag{4}$$

$$+ \mu \sum_{v=1}^{m+1} \sum_{v' \neq v} \beta_{vv'} \mathrm{HSIC}(\mathbf{U}^{\mathrm{v}}, \mathbf{U}^{\mathrm{v}'}), \ \text{ s.t. } \sum \alpha_{\mathrm{v}} = 1, ||\boldsymbol{\beta}_{\mathrm{v}}||_2 = 1, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{F}, \{\mathbf{U}^{\mathrm{v}}\} \geq 0,$$

where λ and μ are nonnegative hyperparameters. It is worth noting that α_v weights the reconstruction error between $\mathbf{X}^v$ and $\mathbf{F}\mathbf{U}^v$, while $\boldsymbol{\beta}_v$ weights the correlation between $\mathbf{U}^v$ and $\mathbf{U}^{v'}$ $(\forall v' \neq v)$. In other words, once $\mathbf{X}^v$ is noisy, α_v will be assigned to a small value as it cannot be recovered well by $\mathbf{F}\mathbf{U}^v$, while $\boldsymbol{\beta}_v$ will take a large value in order to decorrelate $\mathbf{U}^v$ with $\mathbf{U}^{v'}$ $(\forall v' \neq v)$. In this way, NAIL adaptively embeds incomplete views and weak labels into a common latent subspace, and non-linearly decorrelates weight matrices with adaptively weights, enabling to complete missing labels in presence of both noisy views and imbalanced labels. Once (4) is solved, the prediction for missing labels is made by thresholding $\sigma(\mathbf{f}_{i:}^T \mathbf{u}_{:j})$ with a threshold of 0.5.

3 Experiments

3.1 Experimental Settings

We conduct experiments on four benchmark multi-view multi-label datasets: Corel5k[1], Pascal07 (see Footnote 1), Yeast dataset[2] and Emotions[3]. The proposed NAIL[4] is compared with four state-of-the-art methods: lrMMC [7], McWL [9], iMVWL [8] and NAIM^{3}L [5]. lrMMC and McWL are adopted by filling missing features with zero, and iMVWL and NAIM^{3}L are originally designed for incomplete multi-view weak-label learning. NAIL uses the Gaussian kernel in HSIC, and NAIL-L is its variant with the linear kernel.

We tune the hyperparameters of lrMMC, NAIL-L and NAIL on all datasets, and tune the hyperparameters of McWL, iMVWL and NAIM^{3}L on the Yeast and Emotions datasets by grid search to produce the best possible results. On the two image datasets, hyperparameters of McWL, iMVWL and NAIM^{3}L are selected as recommended in the original papers. We select the values of hyperparameters λ and μ from $\{10^i | i = -3, \ldots, 3\}$, and the ratio r_k of $\frac{k}{d}$ from $\{0.2, 0.5, 0.8\}$ for NAIL and NAIL-L. We set $s =$

[1] http://lear.inrialpes.fr/people/guillaumin/data.php.

[2] http://vlado.fmf.uni-lj.si/pub/networks/data/.

[3] http://www.uco.es/kdis/mllresources.

[4] The code and supplement: https://github.com/mtics/NAIL.

$a = 0.5$ and $\gamma = 2$ in experiments. We randomly sample 2000 samples of each image dataset, and use all samples from the Yeast and Emotions datasets in the experiment. We randomly remove $r\%$ samples from each feature view by ensuring that each sample appears in at least one feature view, and randomly remove $s\%$ positive and negative samples for each label. We randomly select 70% of the datasets as the training set and use the rest as the validation set, and repeat this procedure by ten times and report the average values and the standard deviations. The prediction performance is evaluated by two metrics: Hamming Score (HS) [11] and Average Precision (AP) [1]. In this work, our goal is to complete the missing labels in the training set.

3.2 Experimental Results

Table 1. Experimental results on four real-world datasets at $r\% = 50\%$ and $s\% = 50\%$. The best results are highlighted in boldface, and the second best results are underlined.

		lrMMC		McWL		iMVWL		NAIM^{3}L		**NAIL-L**		**NAIL**	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	**Mean**	STD
Emotions	HS	0.5057	0.0125	0.6303	0.0031	0.6281	0.0082	0.6911	0.0068	0.6920	0.0307	**0.7135**	0.0104
	AP	0.5293	0.0140	0.6102	0.0111	0.6006	0.0029	0.6783	0.0149	0.6923	0.0291	**0.7017**	0.0099
Yeast	HS	0.7275	0.0002	0.7420	0.0020	0.7337	0.0113	0.7089	0.0003	**0.7522**	0.0049	0.7462	0.0081
	AP	0.6503	0.0000	0.6936	0.0040	0.7219	0.0037	0.6665	0.0113	**0.7267**	0.0102	0.7235	0.0187
Corel5k	HS	0.9084	0.0089	0.9070	0.0001	0.9581	0.0090	0.9575	0.0174	0.9792	0.0064	**0.9800**	0.0058
	AP	0.1897	0.0021	0.1527	0.0052	0.2643	0.0005	**0.5212**	0.0142	0.3594	0.0834	0.3436	0.0028
Pascal07	HS	0.9194	0.0009	0.8132	0.0004	0.8690	0.0144	0.9211	0.0071	0.9450	0.0131	**0.9480**	0.0096
	AP	0.3998	0.0013	0.3438	0.0032	0.4364	0.0169	0.4494	0.0076	**0.4892**	0.0138	0.4828	0.0188

Evaluation of Comparing Methods. Table 1 shows the experimental results of all comparing methods on four real-world datasets at $r\% = 50\%$ and $s\% = 50\%$. From Table 1, we can see that NAIL and NAIL-L outperform comparing methods in most of the cases. The performance superiority probably comes from their ability on handling noisy views and imbalanced labels, and decorrelating weight matrices for redundancy removal in an adaptive way. The incompleteness of multi-view data causes the performance degradation of lrMMC and McWL. iMVWL and NAIM^{3}L outperform lrMMC and McML in most cases, but perform worse than NAIL and NAIL-L. There are two possible reasons: one is that iMVWL assumes that the label matrix is low-rank, and the other is that both iMVWL and NAIM^{3}L treat multiple views equally. In contrast, NAIL and NAIL-L measure the importance of each view by adaptively choosing appropriate values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. In summary, it shows that once a low-dimensional space indeed contains nonlinear transformations about features and labels, NAIL enables to save their structural properties and uses the HSIC to capture correlations between them.

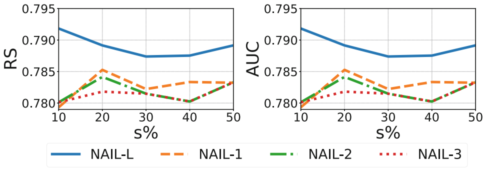


Fig. 2. Ablation study of NAIL on the Corel5k dataset at $r\% = 50\%$ by varying $s\%$ from 10% to 50% by step 10%.

Ablation Study. To investigate the effects of NAIL-L's components, we introduce three variants of NAIL-L, namely NAIL-1, NAIL-2 and NAIL-3. NAIL-1 uses Frobenius norm to measure the reconstruction error of features and labels, instead of $L_{2,1}$ norm and focal loss. NAIL-2 ignores the decorrelation between weight matrices during embedding by simply removing auto-weighted HSIC. NAIL-3 treats multiple views equally in both reconstruction and decorrelation, by omitting $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Figure 2 shows the ablation study of NAIL-L on the Corel5k dataset at $r\% = 50\%$ by varying values of $s\%$. Among the variants, NAIL-3 performs the worst as it fails to detect noisy views. NAIL-1 and NAIL-2 perform worse than NAIL-L, probably because the simple Frobenius norm based loss in NAIL-1 is sensitive to sample outliers and imbalanced labels, and the removal of HSIC in NAIL-2 is harmful for generalization. In contrast, NAIL-L has the best performance in RS and AUC on all datasets, indicating the effectiveness and necessity of its components.

4 Conclusion

In this paper, we propose a novel method called NAIL to deal with incomplete multi-view weak-label data. NAIL jointly embeds incomplete views and weak labels into a shared subspace with adaptive weights, and facilitates the difference between the embedding weight matrices via auto-weighted HSIC. Moreover, to deal with noisy views and imbalanced labels, adaptive $L_{2,1}$ norm and focal loss are used to calculate the reconstruction errors for features and labels, respectively. Empirical evidence verifies that NAIL is flexible enough to handle various real-world problems.

References

1. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: CVPR 2011, pp. 2801–2808. IEEE (2011)
2. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Ann. Oper. Res. **134**(1), 19–67 (2005)
3. Gao, H., Nie, F., Li, X., Huang, H.: Multi-view subspace clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4238–4246 (2015)
4. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005). https://doi.org/10.1007/11564089_7
5. Li, X., Chen, S.: A concise yet effective model for non-aligned incomplete multi-view and missing multi-label learning. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

7. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: Proceedings of the 2013 SIAM International Conference on Data Mining, pp. 252–260. SIAM (2013)
8. Tan, Q., Yu, G., Domeniconi, C., Wang, J., Zhang, Z.: Incomplete multi-view weak-label learning. In: IJCAI, pp. 2703–2709 (2018)
9. Tan, Q., Yu, G., Domeniconi, C., Wang, J., Zhang, Z.: Multi-view weak-label learning based on matrix completion. In: Proceedings of the 2018 SIAM International Conference on Data Mining, pp. 450–458. SIAM (2018)
10. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)
11. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2013)
12. Zhu, C., Miao, D., Zhou, R., Wei, L.: Improved multi-view multi-label learning with incomplete views and labels. In: 2019 International Conference on Data Mining Workshops (ICDMW), pp. 689–696. IEEE (2019)

Natural Language Processing

A Joint Framework with Audio Generation for Rare Gunshot Event Detection

Jun Yin[1,2], Haiyun Du[1], Renjie Wu[1], Ruidong Fang[1], Jucai Lin[1](✉), Yun Huang[1], Weizhen Huang[1], Yapeng Mao[1], and Xiao Luo[2]

[1] Zhejiang Dahua Technology Co. Ltd., Hangzhou, China
{yin_jun,wu_renjie,fang_ruidong,huang_yun1,huang_weizhen,
mao_yapeng}@dahuatech.com, superljcai@163.com
[2] ZheJiang University, Hangzhou, China
bradyluo@zju.edu.cn

Abstract. Nowadays, gunshot detection is closely related to social security, but it needs more attention. The data driven method for gunshot detection which a large corpus of gunshots will be needed for training a neural network is urgently needed. To address this requirement, we propose a novel unified framework, the Gunshot Generation and Detection (GGD), specifically designed for gunshot event detection. It merges an audio generation model with a detection model to partially alleviate the issue of data scarcity problem. Comparative analysis indicates that the GGD model surpasses non-generative models. Remarkably, it outperforms models employing data augmentation techniques. Furthermore, the GGD framework is easy to incorporate with diverse detection network architectures, such as VGGish and Mobile Net. When coupled with a Convolutional Neural Network (CNN), our methodology yields recall varying from 93.98% to 98.20%. These findings demonstrate that this integrated approach significantly enhances the detection performance of gunshot detection models.

Keywords: Gunshot Detection · Audio Generation · Deep Learning

1 Introduction

The issue of public safety is undoubtedly becoming increasingly important for numerous cities across the globe. According to [1], many gunshot incidents are not noticed by emergency organizations. Numerous event detection systems and products rely on video or image. Tuncer et al. [2] propose a real-time system and network architecture for identifying gun violence, which enhances performance of networks well-known in object detection. Gunshot detection technology is designed to note gunfire accident to minimize casualties and losses. While there are areas that cannot be monitored by the video surveillance system, it can be complemented by audio detection, especially gunshot detection.

Gunshot detection is typically tackled using deep learning technique together with data augmentation or designed features, which normally needs a large amount of training

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 133–144, 2024.
https://doi.org/10.1007/978-981-99-7022-3_13

data. Some methods are aimed to solve rare sound event detection on the dataset used in DCASE 2017 task 2. Ding et al. [3] propose an adaptive multi-scale detection(AdaMD) method that processes acoustic events with different temporal and frequency resolutions. To better understand the model's detection capability, Kao et al. [4] investigate the dynamics of LSTM model and nine different pooling methods. They find max pooling on the prediction level gained the best performance. Lydia et al. [5] propose a two-stage pipeline gunshot detection system to detect hunting in the wild with limited data. However, there are limited datasets due to the distinctiveness of gunshots, which is a major obstacle to the advancement of gunshot detection.

Generative models like GANs and diffusion models have remained a research hotspot for a long time. In particular, the diffusion model, once well-known in the field of text-to-image generation [6], has recently been adapted for text-to-audio synthesis domain[7]. AudioLM et al. [8] apply language modeling techniques and gain high-quality audio. Furthermore, the diffusion model [9], based on prompt enhancement, aims to address data scarcity by constructing natural language text that aligns well with audio. It introduces a spectrogram autoencoder for self-supervised representation, rather than modeling through waveform, to ensure effective compression and preservation of speech characteristics.

Due to the data augmentation techniques to enhance gunshot detection models [10] and popularity of diffusion models in image generation [11], we introduce audio generation into gunshot detection. By leveraging audio and corresponding textual descriptions, we generate audio that is similar yet distinct from the original one, thereby increasing the amount of data. Specifically, we establish a strong connection between data generation and detection network. To prevent the impact of poor-quality data, we constrain the proportion of generated data fed into the network through a joint loss function for the two tasks. In this manner, we maximize the use of the limited available data.

We summarize our contributions as three points below. (1) Addressing data scarcity through audio generation: We utilize audio generation techniques to expand the limited gunshots, effectively alleviating data scarcity. (2) Connecting audio generation and gunshot detection models: We establish an effective connection between the audio generation model and the gunshot detection model, forming the gunshot generation and detection(GGD) model. By controlling the quantity and proportion of generated data fed into the network through conditional constraints, we ensure the training quality of the detection model. (3) Superior performance compared to models without GGD: Our approach demonstrates improved performance in gunshot detection when compared to models without audio generation and even models with augmentation.

2 Related works

Gunshot detection is normally solved by either more designed features, data augmentation or various networks. Baliram Singh et al. [12] focus on the analysis of feature importance about gunshot and gunshot-like sounds, which applies machine learning methods like random forest mean and the SHapley Additive exPlanations. Arslan et al. [13] propose a gunshot recognition system containing impulsive sound detection based on frequency information and MFCC. Furthermore, Bajzik et al. [14] propose new features such as spectrogram, MFCC and self-similarity matrix, thereby training the CNN network for gunshot detection.

Though these methods can detect gunshot in a simple but effective way, their performance is severely obstructed by the scarcity of gunshot datasets. Therefore, it's essential to incorporate data augmentation approaches. The ICRCN [15] applies random background mixing, random time shift and random Gaussian noise addition to expand gunshots. The Dos et al. [16] investigate the impact of noise-addition on gunshot detection system, which shows that the noise-addition may affect feature, and decrease detection performance. Another typical way is collecting more data, which crawls gunshot from widely used video websites like YouTube, Tiktok and IMDB [2]. Besides, Rahul et al. [17] record the audio clips in residential areas and at a gun range. Busse et al. [18] create relatively limited gunshots in a physical way, which rely on other kinds of data to generate gunshot-like sound. Park et al. [19] gather gunshots in game to perform gunshot detection in reality, which are similar to real sounds to some extent. The method perform well on gunshots data in UrbanSound8K. However, the cost of these methods and the diversity of gunshots require further improvements. Thus, we propose an effective and controllable GGD model that enriches gunshots by generation.

3 Methods

3.1 GGD Architecture

Data generation can enhance the robustness and generalization of models. To address the challenge of scarcity in high-quality, authentic audio data for gunshot detection tasks, we introduce a strategy that incorporates audio generation into the model training process. This integration culminates in our proposed joint framework GGD, which has online and offline modes corresponding to Figs. 1 and 2 respectively. In the online mode, Mel spectrograms are generated are utilized as partial input, and the parameters of both the generation and detection models are simultaneously updated. The aim is to empower the generation model to produce a large volume of quality superior-quality data during training the detection model. In the offline mode, the parameters of the generation model are frozen which generates features from different textual contents to expand the gunshot category.

The overall online mode operational flow is depicted in Fig. 1. For illustrative simplicity, the figure provides a reduced schematic of a CNN. The GGD input comprises three components: audio of all classes, category labels, and textual descriptions of gunshots. Regarding the generation module, it requires gunshot audio and textual descriptions. The output of the generation module is the spectrogram feature of the gunshot. Within the generation model, the spectrogram is sent into a Self-Supervised Audio Spectrogram Transformer (SSAST) encoder [20]. The generated image is constrained by the text description. The spectrogram, produced by diffusion, denoising, and SSAST decoder, undergoes a data selection module to determine the proportion incorporated into the detection model along with original data. Then the generated features, in conjunction with the features from original data, are directly transmitted to the detection model, thereby extending the amount of available gunshots. Other categories of data are also sent together into the detection model. Notably, this architecture can be adapted for other sound event detection tasks.

The offline mode flow is depicted in Fig. 2, where the parameters are frozen and the generation model is only used for generating features from gunshot text contents. The features generated are sent into detection model directly along with features extracted from original gunshots and other categories. The offline mode saves more time for training and resource consumption compared to online mode.

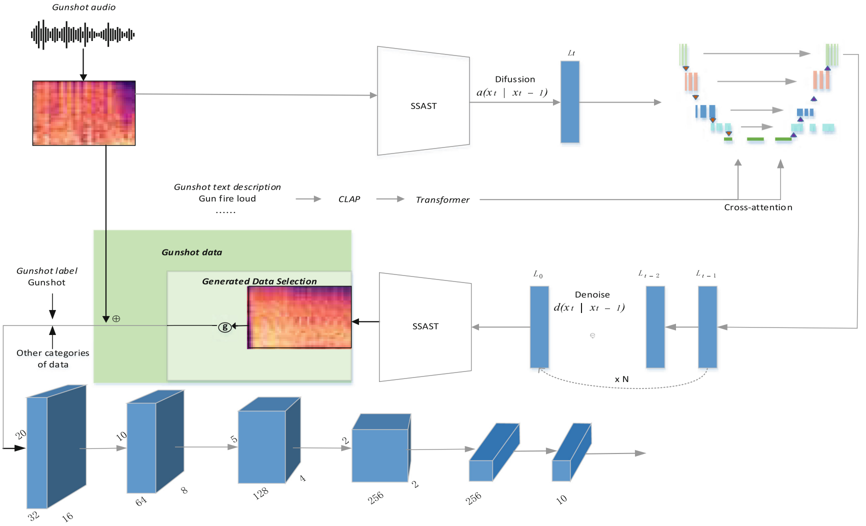


Fig. 1. GGD architecture. Mel spectrogram is obtained from gunshots and fed into SSAST encoder. The text description constrains the generated feature. The feature is generated by diffusion, denoising and SSAST decoder, then goes through data selection module. It determines the proportion of the generated spectrograms that will be integrated with original data in the detection model. Within the data selection module, the output of the gated function, denoted by 'g', signifies the proportionality factor.

3.2 Audio Generation

Generation Module. During audio generation, textual descriptions and gunshot audio are fed into this module. The textual description of gunshot is artificially created. After extracting spectrogram, generation progress starts, which is similar to diffusion model [11]. The first step is to make use of SSAST encoder to characterize the feature. Next the representation vector is sent into diffusion model, which input consists of original data, text embedding, random noise embedding and time step embedding. The output of diffusion model is an image with similar feature corresponding to the text. The Diffusion model consists of encoder and decoder module. The encoder compresses origin data into lower dimension data i.e. latent data, which is called diffusion procedure. The decoder is responsible for restoring the latent data to original data, which is called denoising, which means reverses diffusion procedure.

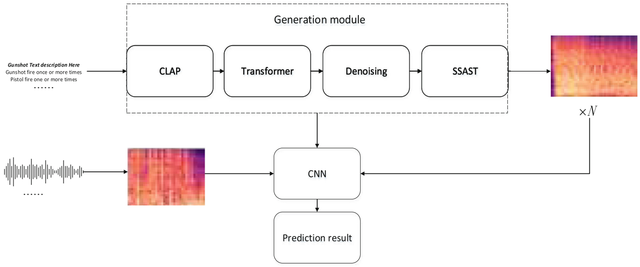

Generation module
Gunshot Text description Here
Gunshot fire once or more times
Pistol fire one or more times
......
CLAP
Transformer
Denoising
SSAST
×N
......
CNN
Prediction result

Fig. 2. Offline mode of GGD structure for gunshot detection.

After t diffusion processes, latent data x_t is obtained. Denoising process means the transition from x_t to x_{t-1} until x_0, which is implemented by U-Net [21]. Denoising process [22] from latent x_t to x_{t-1} can be simply represented in (1), where α_t and σ_t are hyper-parameters and Z represents random Gaussian noise. The first addition term represents the latent distribution predicted by U-Net, where $\epsilon_\theta(x_t, t)$ represents the predicted distribution of noise by U-Net. The difference of x_t and noise distribution is scaled by $\frac{1}{\sqrt{\alpha_t}}$. Then we get the preliminary distribution of latent x_{t-1}. If the second addition term is absent, the predicted latent is prone to overfitting during training, which means the same x_{t-1} value may appear several times. Thus, the additional noise is added for introducing more diversity to generated latent in our generation module. We can get more diverse gunshots feature after t denoising procedures in this way. It can further improve the effectiveness of the generation module and create more diverse inputs for gunshot detection model.

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}}\epsilon_\theta(x_t, t)\right) + \sigma_t Z \tag{1}$$

We take some real and generated spectrograms of gunshots for example to show diversity intuitively between them in Fig. 3. The spectrograms of generated gunshots and real gunshots look similar. While there are differences in background noise, intensity, number of shots and durations. We can qualitatively observe that generated features own rich diversity. It is obviously shown from the Fig. 3 that Audio Generation method enriches the realistic diversity with limited gunshots, which strengthen the generalization of gunshot detection model. It's not just a random reshuffling of the same statistic data. Meanwhile, the gunshot detection model type is less constrained due to generated data, which means less cost for network structure design.

To achieve multimodal conversion from text to speech and ensure that the text and final audio are closely matched, a constraint on the latent space should be applied using the text. Firstly, the CLAP [23]. Text encoder depicts the textual description. The text embedding feature representation is obtained by following transformer structure and then is used to compute cross-attention matrix to constrain the model output image, ensuring the data and text are related. After N denoising processes, x_0 is calculated. The SSAST decoder then decodes the data to get generated spectrogram. We simplify this process by removing the vocoder of Make-An-Audio and directly feeding generated feature and original feature into network without saving audio files.

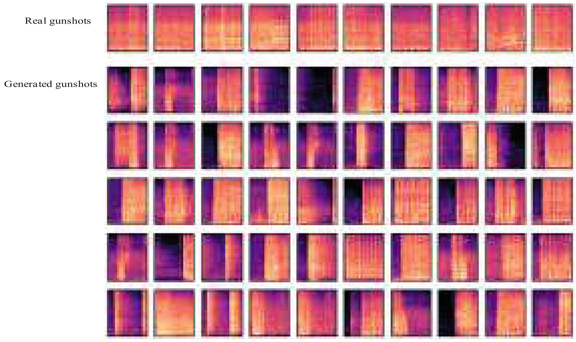


Fig. 3. Mel Spectrograms of real gunshots and generated gunshots. The row 1 represents real gunshots, the remaining spectrograms represent generated gunshots based on real gunshots.

Loss Function. The loss function is the sum of the two losses. The formula is represented as Eqs. (2–4) below, where L denotes loss for GGD frame, L_1 for audio generation module, L_2 for CNN network, ϵ_θ represents the calculated noise in denoising process and ϵ for the diffusion in (3). The t respectively represents every diffusion and denoising procedure, which is a random value determined by a hyper-parameter and text is the input textual description. In (4), y_{ic} denotes sign function. If sample i belongs to class c, y_{ic} equals 1 otherwise 0. p_{ic} Denotes the inference probability belonging to class c. Cross entropy loss function is used in detection network to minimize the gap between label and prediction to optimize model.

$$L = L_1 + L_2 \tag{2}$$

$$L_1 = \|\epsilon_\theta(x_t, t, text) - \epsilon\|_2^2 \tag{3}$$

$$L_2 = -\frac{1}{N}\sum\nolimits_i \sum\nolimits_{c=1}^{10} y_{ic}\log(p_{ic}) \tag{4}$$

Data Selection for Training. During training, not all generated data is fed into the network due to noise and poor quality of generated data. The proportion of added data changes gradually throughout the training process according to loss value of L_1. $L_{1,i}$ denotes the loss of generation module at epoch i. When $L_{1,i}$ becomes less than 20% of $L_{1,0}$, which is the loss value of the first epoch, we set E_s as i as Eq. (5). The quality of generated data at this time is better compared to the first epoch. The proportion equals 0.05 as shown in (6) and the generated data is added into network.

$$E_s = i, when L_{1,i} < 0.2 * L_{1,0} \tag{5}$$

$$g = \begin{cases} 0, i < E_s \\ \min\left(0.2, \lfloor \frac{i-E_s}{15} + 1 \rfloor * 0.05\right), i \geq E_s \end{cases} \tag{6}$$

This upper bound of proportion is 0.2 and 15 are hyper-parameters, which are determined by multiple experiments and a trade-off choice between performance and time cost. Subsequently, the proportion is adjusted every 15 epochs, with each adjustment increasing the quantity by 0.05 until reaches bound. The method of data selection is random sampling. When the proportion of data is confirmed, we select data without replacement until the quantity meets requirements. In this way, the diversity and quality of dataset have both been enhanced, which results in improving the capability of detection model.

3.3 Offline Mode for GGD

Figure 2 illustrates the implementation of the GGD framework in an offline mode, where the generation module operates independently from CNN. Specifically, the generation module can be frozen, indicating its exclusive use for the generation of spectrograms, without concurrent training with the CNN. After the training and parameter fixation of the generation network, this module models the features corresponding to provided textual contents. It is noteworthy that the diversity in textual content engenders variance in the derived features, substantially augmenting data diversity.

The offline mode facilitates the segregation of the generation model from the CNN, circumventing the need for time-consuming, simultaneous online training, thereby reducing training duration and resource utilization. Subsequent stages adhere to the process established in online training, whereby data generated by the generation model is amalgamated with the original dataset, and then relayed to the neural network. This process elucidates offline data generation for gunshot event detection.

4 Experiments and Analysis

4.1 Dataset

Our benchmark dataset consists of two parts, which are used to train and evaluate. The training dataset encompasses 10 distinct audio categories, including other sound categories that could potentially generate false alarms. The data is derived from a combination of publicly available datasets and our recordings. The public datasets we utilized contain AudioSet, Dcase2017, Dcase2018 and Dcase2020, while self-recorded data categories feature speech, footstep, and others. In the case of gunshots, which is inherently challenging to acquire, we have supplemented the data obtained not only from public datasets but also from shooter games or videos. This approach has enabled us to compile an extensive set of WAV format audio files, totaling approximately 22,000 original audio files. These audio files were resampled with 16kHz and cut into 1s length segments. For the length of data less than 1s, we extended these files into 1s with zero. By imposing a maximum limit of 10,000 samples per category, we ensured a balanced distribution across all categories. As a result, we obtained approximately 75,000 audio clips training our benchmark model. A comprehensive overview of our data can be found in Table 1 below.

Our benchmark test dataset is derived from the publicly accessible and authentic, real-world recorded dataset, the Gunshot Audio Forensics Dataset [24]. This invaluable dataset is a subset of the comprehensive firearms audio forensics dataset funded by NIJ Grant 2016-DN-BX-0183. It encompasses a diverse range of gun types, recording devices, various directions and distances. There are about 6,400 gunshot samples for the models' performance evaluation.

In summary, our data is methodically partitioned into three distinct components: the training set, the validation set, and the test set. The training and validation set share a common origin, with a fixed allocation of 1,000 samples per category designated for the validation set. Meanwhile, the test set comprises the aforementioned 6,400 samples from the Gunshot Audio Forensics Dataset.

Table 1. Overview of the dataset.

Class	Percentage(%)
Gunshot	13.3
Speech	13.3
Clap	8.3
Footstep	9.5
Door	6.7
Glass break	4.2
Dog	13.3
Engine	13.3
Drilling	9.8
Honk car	8.1

4.2 Implementation Details

In this study, Mel spectrograms are extracted and then sent to GGD framework. They are extracted using the Torchaudio toolkit. The number of Mel filters is set to 32. Three network structures are carried out on a range of experiments consisting of CNN, VGGish [25] and MobileNet [26]. We performed experiments utilizing both benchmark datasets and generated datasets for a robust analysis of performance. Each of the models was subjected to a rigorous training process with 300 epochs. We employed Adam optimizer with an initial learning rate of 0.001.

We experimented with various upper limits for adding the proportion of generated data, including 50%, 20%, 16.7% and 10%. By testing various proportions, we were able to identify the most suitable ratio of generated data to be added. This careful implementation of generated data contributes to the development of more accurate and reliable gunshot detection systems.

4.3 Experiments Results

In this investigation, we utilized deep learning architectures to train models on our dataset, subsequently assessing performance on test set to calculate recall rate. Table 2 delineates the results of models trained with and without synthesized data. The results reveal that the inclusion of synthesized data typically enhances the recall rate of gunshot detection across the three models. The results imply that supplementing models with generated data can bolster their performance to a certain degree. Taking the CNN network as an example, we observed that when the proportion of generated data added is 20%, the result increases from 93.98% to 96.76%. When the proportion becomes 50%, the result reaches an impressive 98.20%. What's more, the GGD framework is suitable for various network. The performance of GGD with Mobile Net and VGGish is superior to the models without audio generation.

We also conducted experiments to validate the effects of reducing training dataset size from 10,000 to 8,000. The results are shown in Table 3. The overall result is lower than when the original training dataset size is 10,000, but it's also evident the addition of generated data improves the model's performance. Results have shown that GGD can achieve good results on data from different sources and even with a smaller training set. We can ensure that the model remains balanced and effective across various data categories by carefully selecting the optimal proportion of generated data.

Table 2. The experimental results for adding various proportions of data.

Network architecture	Proportion of generation data	Recall rate
CNN	0	93.98%
	1/2	98.20%
	1/5	96.76%
	1/6	97.11%
	1/10	97.66%
Mobile Net	0	66.72%
	1/2	97.55%
	1/5	97.31%
	1/6	81.04%
	1/10	67.74%
VGGish	0	94.15%
	1/2	98.47%
	1/5	95.56%
	1/6	97.81%
	1/10	97.16%

Table 3. The result of reducing the number of training sets.

Network architecture	Proportion of generation data	Recall rate
CNN	0	90.59%
	1/2	93.09%
	1/5	94.51%
	1/6	93.14%
	1/10	92.61%
Mobile Net	0	61.05%
	1/2	76.34%
	1/5	90.20%
	1/6	80.74%
	1/10	81.23%
VGGish	0	95.28%
	1/2	96.53%
	1/5	98.03%
	1/6	96.56%
	1/10	98.34%

Data generation can be considered a data augmentation method. To demonstrate the feasibility of data generation, we compared it to augmentation using the CNN detection network. The augmentation includes noise addition, pitch-shifting, reverberation, and spectrogram mask. The results are presented in Table 4, where the result of generated data in training phase was the lowest recall rate in Table 2. The results indicate that without applying augmentation techniques, the recall rate is 93.98%. Adding different augmentation techniques has varying effects on the results, with noise addition and pitch shifting achieving the closest results to GGD.

Spectrogram mask get a lower 78.82% recall rate than no augment, which is usually added for imitating channel loss in network transmission and does not exist in test dataset. That may bring negative impact to results. Reverberation method get a lower 81.38% recall rate than no augment, which is usually added for imitating room impulse response, while our test dataset is recorded in wild absent of obvious room impulse response. Pitch shifting could change the frequency component of audio, which draws diversity into data but may also affect feature. If the added noise mismatches noise in test dataset or its intensity is improper, it may bring impact to detection system. Our method generates more gunshots without affecting original features and brings more diversity, which result in a 96.76% advantage over other methods. This experiment further demonstrates that GGD offers significant advantages over data augmentation techniques. Furthermore, integrating generated data and original data into the training process can enhance the model's robustness and generalization capabilities, ultimately improving the detection ability of models on real data.

Table 4. Comparison between our method and augment method.

Augment method	Recall rate
No augment	93.98%
Pitch-shifting	94.28%
Noise addition	96.17%
Reverberation	81.38%
Spectrogram mask	78.82%
Ours	96.76%

5 Conclusion

We focused on the detection of gunshots. To tackle the data scarcity issue, we proposed GGD, that jointly trains generation and detection networks. We can obtain a larger number of gunshots in this way, thereby mitigating the shortcomings of inadequate detection capabilities and weak generalization abilities in detection networks caused by data scarcity. The framework GGD successfully combines generation and detection models which outperformed those methods that only consist of detection models, even with data augmentation in gunshot detection. It can also be considered for application in other detection tasks. In the future, we may explore better integration of data augmentation methods and further optimize our model to achieve a more lightweight generation model component. Besides it's possible to consider combining it with the physical generation of data to generate possibly more realistic gunshots. These advancements will pave the way for more efficient and accurate detection systems, capable of handling a wide range of rare sounds and contributing to the development of safer environments.

References

1. Irvin-Erickson, Y., Bai, B., et al.: The effect of gun violence on local economies. Urban Institute, Washington, DC (2016)
2. Tuncer, T., Dogan, S., Akbal, E., et al.: An automated gunshot audio classification method based on finger pattern feature generator and iterative relieff feature selector. Adıyaman Üniversitesi Mühendislik Bilim. Derg. **8**, 225–243 (2021)
3. Ding, W., He, L.: Adaptive multi-scale detection of acoustic events. IEEE/ACM Trans. Audio, Speech Lang. Proc. **28**, 294–306 (2020)
4. Kao, C.C., Sun, M., Wang W., et al.: A comparison of pooling methods on lstm models for rare acoustic event classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020)
5. Katsis, L.K., Hill, A.P., et al.: Automated detection of gunshots in tropical forests using convolutional neural networks. Ecol. Ind. **141**, 109128 (2022)
6. Nichol, A., Dhariwal, P., Ramesh, A., et al.: GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Model. arXiv.2112.10741. (2021)
7. Yang, D., et al.: Diffsound: discrete diffusion model for text-to-sound generation. IEEE/ACM Trans, Audio Speech Lang. Proc. **31**, 1720–1733 (2023)

8. Borsos, Z., Marinier, R., Vincent, D., et al.: AudioLM: a Language Modeling Approach to Audio Generation, arXiv. 2209.03143 (2022)
9. Huang, R., Huang, J., Yang, D., et al.: Make-An-Audio: Text-To-Audio Generation with Prompt-Enhanced Diffusion Models, arXiv. 2301.12661. (2023)
10. Alex, M., Lauren, O., Gabe, M., Ryan, H., Bruce, W., George, M.: Low cost gunshot detection using deep learning on the Raspberry Pi. In: IEEE Conference Proceedings (2019)
11. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
12. Singh, R.B., Zhuang, H.: Measurements, analysis, classification, and detection of gunshot and gunshot-like sounds. Sensors **22**(23), 9170 (2022)
13. Arslan, Y.: Impulsive sound detection by a novel energy formula and its usage for gunshot recognition. arXiv preprint arXiv:1706.08759, (2017)
14. Bajzik, J., Prinosil, J., Koniar, D.: Gunshot detection using convolutional neural networks. In: 2020 24th International Conference Electronics, pp. 1–5. IEEE (2020)
15. Bajzik, J., Prinosil, J., Jarina, R., Mekyska, J.: Independent channel residual convolutional network for gunshot detection. Inter. J. Adv. Comput. Sci. Appli. (IJACSA) **13**(4) (2022)
16. Dos Santos, R., Kassetty, A., Nilizadeh, S.: Disrupting audio event detection deep neural networks with white noise. Technologies **64** (2021)
17. Nijhawan, R., Ansari, S.A., Kumar, S., et al.: Gun identification from gunshot audios for secure public places using transformer learning. Sci. Rep. **12**(1), 13300 (2022)
18. Busse, C., et al.: Improved gunshot classification by using artificial data. In: 2019 AES International Conference on Audio Forensics (2019)
19. Park, J., et al.: Enemy Spotted: in-game gun sound dataset for gunshot classification and localization. In: 2022 IEEE Conference on Games, pp. 56–63 (2022)
20. Gong, Y., Lai, C.-I., Chung, Y.-A., Glass, J.: Ssast: selfsupervised audio spectrogram transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 10699–10709 (2022)
21. Olaf Ronneberger: U-Net: convolutional networks for biomedical image segmentation. In: Nassir Navab (ed.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III, pp. 234–241. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Jain, J., et al.: Denoising diffusion probabilistic models: HO. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
23. Elizalde, B., Deshmukh, S., Ismail, M., Wang, H.: CLAP: Learning Audio Concepts From Natural Language Supervision (2022)
24. Gunshot Audio Forensics Dataset (2017). http://cadreforensics.com/audio/,
25. Hershey, S., Chaudhuri, S., Ellis, D. P.W., et.al.: CNN architectures for large-scale audio classification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 131–135. (2017)
26. Howard, A., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. In: Computer Vision and Pattern Recognition (2017)

Ancient Chinese Machine Reading Comprehension Exception Question Dataset with a Non-trivial Model

Dongning Rao[1], Guanju Huang[1], and Zhihua Jiang[2](✉)

[1] School of Computer, Guangdong University of Technology,
Guangzhou 510006, People's Republic of China
raodn@gdut.edu.cn, 2112105150@mail2.gdut.edu.cn
[2] Department of Computer Science, Jinan University,
Guangzhou 510632, People's Republic of China
tjiangzhh@jnu.edu.cn

Abstract. Ancient Chinese Reading Comprehension (ACRC) is challenging for the absence of datasets and the difficulty of understanding ancient languages. Further, among ACRC, entire-span-regarded (Entire spaN regarDed, END) questions are especially exhausting because of the input-length limitation of seminal BERTs, which solve modern-language reading comprehension expeditiously. To alleviate the datasets absence issue, this paper builds a new dataset ACRE (Ancient Chinese Reading-comprehension End-question). To tackle long inputs, this paper proposes a non-trivial model which is based on the convolution of multiple encoders that are BERT decedents, named EVERGREEN (EVidence-first bERt encodinG with entiRE-tExt coNvolution). Besides proving the effectiveness of encoding compressing via convolution, our experimental results also show that, for ACRC, first, neither pre-trained AC language models nor long-text-oriented transformers realize its value; second, the top evidence sentence along with distributed sentences are better than top-n evidence sentences as inputs of EVERGREEN; third, comparing with its variants, including dynamic convolution and multi-scale convolution, classical convolution is the best.

Keywords: Ancient Chinese · Reading comprehension · Dataset · Evidence extraction · Convolution

1 Introduction

To preserve ancient Chinese culture, where Ancient Chinese (AC) is a pivotal carrier, natural language processing (NLP) on AC is an interesting and imminent [15] task, where the lacking of linguistic resources harden the difficulty of AC understanding tasks, including AC reading comprehension (RC). Among ACRC tasks, entire-text-span regarded (Entire-text-spaN regarDed, END) questions are especially complicated because they promote the difficulty of learning syntactic features for NLP models from sentence-level to document-level and cannot be answered based on the first few tokens of the questions or word matching [10]. Unfortunately, to the best of our knowledge, there are no available ACRC datasets yet, especially for END questions.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 145–158, 2024.
https://doi.org/10.1007/978-981-99-7022-3_14

To alleviate this situation, we built a new dataset, named ACRE (Ancient Chinese Reading-comprehension End-question) along with a new model, which is called EVERGREEN (EVidence-first bERt encodinG with entiRE-tExt coNvolution). ACRE comprises of 4100 AC passages with END questions, which are manually collected. EVERGREEN has multiple pre-trained language models (PLMs) as encoders for splitting input parts and a convolution component which condenses an entire-text encoding.[1]

Besides building the first ACRC dataset, our experiments showed that:

1. PLMs trained with similar languages that have abundant resources like modern Chinese are better than PLMs trained with target low-resource languages (e.g., AC) [13] more often than not.
2. Length-text oriented PLMs cannot outperform customized ensemble models.
3. When we compress the encoding of long text via CNN, classical CNN beats its variants, such as dynamic convolution (DCN) [19] and multi-scale convolution (MsCNN) [14].
4. Evidence extraction, which has been proved to be successful in RC [18], should be combined with the sentences distribution strategy to boost the prediction accuracy.

The rest of this paper is organized as follows. Related work is briefly written in the next section. Then, the architect of EVERGREEN is proposed after introducing ACRE and the problem. At last, experiments are followed by a conclusion section.

2 Related Work

2.1 Ancient Chinese Reading Comprehension

As a sub-task of RC, which is a principal task of natural language understanding that keeps attracting attentions from researchers [18], ACRC aims to automatically answer questions according to an ancient Chinese passage. RC in language other than English can also be viewed as new tasks, because the processing of difference language might be varied widely in many aspects, including segmentation, part-of-speech tagging, and syntactic analysis [13]. E.g., the prominent Chinese writer and educator, Ye Shengtao, believed that the three key differences between modern and ancient Chinese are: first, from the aspect of vocabulary, ancient Chinese used one-character words which could not be used along nowadays; second, from the aspect of grammar, many unambiguous words in modern Chinese have multiple explanations in ancient Chinese; third, from the aspect of function words, many ancient Chinese words vanished.

The difficulty of ACRC lies in its exploratory and comprehensive for the inference and deduction requirements[2]. Because the above-mentioned differences between modern and ancient Chinese make ACRC harder than modern Chinese RC, ACRC had been employed to measure the level of mastering the Chinese in almost all formal examination for native speakers in China [15], including the destiny-decided and 10-million-students-involved Gaokao. Questions in ACRC can be categorized into at least four

[1] Both ACRE and the source code of EVERGREEN will be released on GitHub after publication.

[2] See eea.gd.gov.cn.

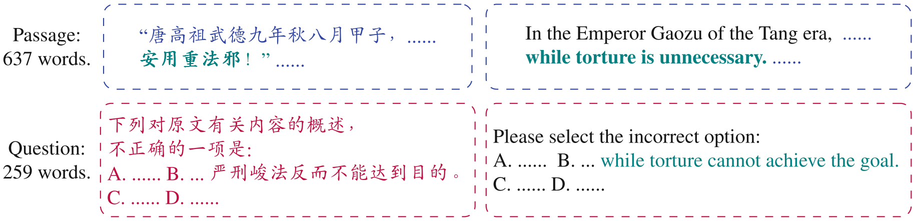

Passage:
637 words.
“唐高祖武德九年秋八月甲子，......
安用重法邪！”......
In the Emperor Gaozu of the Tang era, ...
while torture is unnecessary. ......
Question:
259 words.
下列对原文有关内容的概述，
不正确的一项是：
A. ...... B. ... 严刑峻法反而不能达到目的。
C. ...... D. ......
Please select the incorrect option:
A. ...... B. ... while torture cannot achieve the goal.
C. ...... D. ......

Fig. 1. A Minimum ACRC END Question Illustration.

types: ancient-modern Chinese translation, sentence segmentation, span extraction and END questions [18], where END questions are often placed at the end of ACRC section and further promote the difficulty of learning syntactic features for NLP models from sentence-level to document-level [12].

Figure 1 is an ACRC END question in the 2021 Gaokao, where the article has 637 Chinese characters, and the question is an entire-passage-regarding question whose length is 259. In Fig. 1, the blue excerpted article passage is in ancient Chinese, and purple sentences are the question with four options in modern Chinese. Further, words as crucial evidence for this problem are colored in teal, and we provide the English translation of key sentences in black along with the question and options.

Because ancient Chinese is a low-resource language, for ACRC, building datasets is one of our top priority duties. There are over 50 English RC datasets, including RACE [5], which is collected from the English exams for the Chinese students, ReClor [16], whose questions are obtained from the Law School Admission Council, and AdversarialQA [1]. By contrast, many low-resource language RC dataset are only recently built [8]. Further, while challenging datasets are more interesting, most existing Chinese RC datasets are in modern Chinese and collected from primary examinations which are designed for Chinese-as-a-second-language students, e.g., C^3 [11] and GCRC [12]. As far as we know, only Native Chinese Reader (NCR) [15] provides 1125 ACRC passages.

2.2 Reading Comprehension via Pre-trained Language Models

Most state-of-the-art (SoTA) models for RC are based on PLMs. E.g., the SoTA approach on Race is an ensemble ALBERT-xxlarge [4]; the best model for ReClor is ALBERT [6]; the first choice for AdversarialQA is RoBERTa [1]; (ensembled) BERT is the SoTA model for and C^3 [17] and GCRC [12]. While our one eye is on the fruition of BERT and the other eye on the shift between modern Chinese and ancient Chinese [13], a few ancient Chinese PLMs were proposed. AnchiBERT [13] and GuwenBert[3] are the only two available models, as far as we know. AnchiBERT is based on BERT and trained with its self-built ancient Chinese corpora, and focused on poem classification, ancient-modern Chinese translation, poem generation, and couplet generation.

[3] The model of GuwenBERT is available on the github.

GuwenBERT is based on RoBERTa [7] and trained with ancient literature of Shuzhige[4], and focused on sentence segmentation, punctuation, and named entity recognition.

However, the 512 tokens input length limitation of BERT is the Achilles' heel of PLMs like BERT, e.g., both AnchiBERT and GuwenBERT have the input-length-limitation as BERT, and therefore for tasks like entire-text-span regarded questions, truncation is unavoidable. To overcome the length limitation, previous studies proposed four solutions: first, truncate the passage; second, extract evidences and feed it into BERTs [18]; third, ensembleing multi-BERT, including straightforward voting, which decides according to the majority vote, and stacking; and fourth, facilitating PLMs whose length limitation is bigger than 512 token, including T5 [9] and Longformer [2].

Chunking long text into segments and inputting them to different branches is the step stone of BERTs based solutions. However, for END questions, all encoding are to be put together to remedy failures in extracting processes, eventually. Leveraging the recurrent mechanism for cross-segments-attentions can put all information together, but convolution has proven to be more successful in computer vision tasks. E.g., DCN networks [19], which adds 2D offsets to the regular grid sampling locations in the standard convolution along with deformable RoI pooling, is a previous SoTA approach for the Microsoft common objects in context dataset. As another example, MsCNN that adaptively selects multi-scale features in a CNN model also leads to better results [14].

3 ACRE

This paper collected the first ancient Chinese reading comprehension entire-text-span regarded question dataset, which is called ACRE. In this section, we will discuss the collecting procedure with the statistics and biases of ACRE after specifying our task.

3.1 Task Specification

ACRC END question are questions that have four options, among which exactly one is the answer. These questions are exceptional-questions that are based on the summarization of an AC passage, but the question itself is expressed in modern Chinese. This bi-language setting is risen because, although AC recorded historic decisions that reshaped our world everlastingly, people only use modern Chinese nowadays. Therefore, ACRE items are only available in papers from different examinations.

3.2 Statistics of ACRE

ACRE has 2975 passages (with END questions) from websites and 1125 passages from NCR. We study the length of ACRE, in which over 99.85% items are shorter than 1536, and most of them are select-false questions, see Table 1.

[4] Shuzhige.

Table 1. Length Statistics of Items in ACRE.

	# tokens			
	≤ 512	≤ 1024	≤ 1536	total
# items	275	4029	4094	4100
ratio	0.67%	98.27%	99.85%	100%

3.3 Data Collecting

ACRE is collected from the web with data cleaning and pre-processes, which include duplication eliminating, translation, function words erasing and data argumentation.

Data Source. Following suggestions from previous studies [3], ACRE is manually collected from publicly available legal educational resources websites, and we did not use any spider. In ACRE, all questions are proposed and answered by experts and prepared for public exams, which are not protected by the Copyright Law. The source Websites are listed in Table 2.

Table 2. Source Web Site of ACRE.

#	url
1	http://yinruiwen.com
2	http://www.5156edu.com
3	https://www.wenyiso.com
4	http://www.yuwen360.com
5	http://m.cyyangqiguan.com
6	https://yuwen.chazidian.com

Removing duplication and adjusting label distributions on those raw data from different web sources are the major tasks of the data cleaning operation. At last, we merge all ACRC END questions in NCR[5] into ACRE.

Translation. As a remedy for the bi-language setting, we use Bing for the translation. As shown in Fig. 1, the passage in ACRE is in AC, while the question and options are in modern Chinese. However, while this setting might alleviate the reading burden for the students, previous evidence-extraction-based RC approaches prefer all passages, question, and options in the same language. Therefore, as the translation between AC and modern Chinese is available but unsteady, we append modern Chinese translations to the passage and attached AC translations to the options and questions to ACRE. Figure 2 provides an example of translating question and options into AC.

[5] https://sites.google.com/view/native-chinese-reader/.

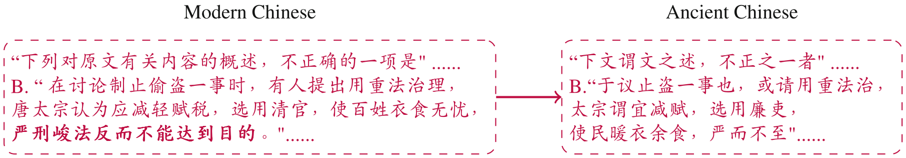

Modern Chinese
Ancient Chinese
“下列对原文有关内容的概述，不正确的一项是" ......
B. “在讨论制止偷盗一事时，有人提出用重法治理，
唐太宗认为应减轻赋税，选用清官，使百姓衣食无忧，
严刑峻法反而不能达到目的。"......
“下文谓文之述，不正之一者" ......
B.“于议止盗一事也，或请用重法治，
太宗谓宜减赋，选用廉吏，
使民暖衣余食，严而不至"......

Fig. 2. Example of Question & Options Translation.

Function Words Erasing. A unique feature of AC is the function word, which can be erased. The vanished function words (e.g., pronouns, adverbs, prepositions, conjunctions, auxiliary words, and exclamations) of ancient Chinese signaled grammatical relationships but have no lexical meaning. Therefore, erasing vanished function words appears attractive because the erasing can shorten the input length. See Fig. 3 for an illustration.

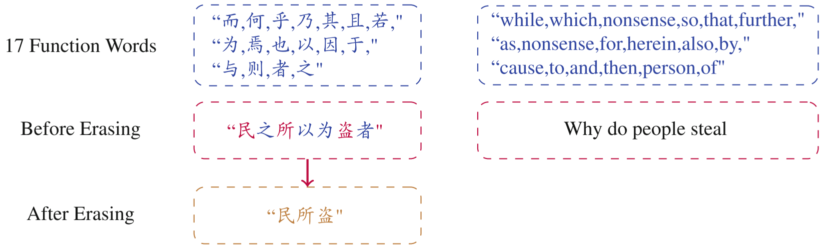


Fig. 3. Example of Function Word Erasing.

3.4 Data Biases and Challenges

ACRE is collected from exams devised by experts in AC, which induces four biases.

1. END questions are often opinions where experts can make mistakes.
2. ACRE items are designed for test papers, therefore the length is limited, as longer passage will take longer time for students.
3. Some questions are extraordinarily sophisticated because they are prepared for tests like Gaokao, which is one of the toughest selective exams in the world. Hence, the hardness of those questions will cause a very low accuracy rate.
4. Most passages are written before the Ming dynasty.[6]

4 EVERGREEN

We proposed EVERGREEN in this section after formally defining our problem.

[6] Thanks to the anonymous NeurIPS reviewer. Although we can draw a line at the Chinese renaissance around 1920 as the boundary between ancient and modern Chinese, fictions which are written in or after the Ming dynasty are not in this scope.

4.1 Problem Formalization

For ACRE problems, there is a 3-tuple $< D, Q, A >$ with an answer label L, where D is a set of passages, Q is a question, and A is the set of four options, and L is the label which indicates the correct answer. We look for a prediction $\hat{L}$ whose conditional probability is maximal among all answer candidates $L' = l_1, l_2, l_3, l_4$ given $< D, Q, A >$.

$$\hat{L} = \underset{L'}{\operatorname{argmax}} P(L'|D, Q, A) \tag{1}$$

4.2 Overview of Network Architecture

The architect of EVERGREEN is as Fig. 4, which leverages multi-BERT as a base encoder and a convolution layer to fit the encoding into a fixed-length, flattened layer. First, inputs of the model include question, options, and passage, where we reduce question into one token to save space because the question is either "select the only correct option among four options" or "select the only wrong option among four options". Second, every option, along with sentences that are evidences for this option, are encoded by a PLM. We illustrate four branches at the bottom of Fig. 4. Third, because encoding from four PLMs is still too long for a transformer layer, we facilitate a convolution layer and pool them into a sequence of proper-size tokens. Fourth, a fully connected layer attached to a transformer encoder layer predicts answers with a Softmax function.

4.3 Formalized Procedure

Let the passage be $D =< s_1, ..., s_{|D|} >$, all options are $A =< a_1, a_2, a_3, a_4 >$, and the question is Q. $S = \{D \cup Q \cup A\} = \{s_1, s_2, ...s_{|S|}\}$ are the set of sentences, where $s_m =< c_1, c_2, ...c_{|s_m|} >, 0 < m \leq |S|$ is a sequence of $|s_m|$ tokens. Then, EVERGREEN has four word-level encoders, each has a question token q, an option a_i, $(0 \leq i \leq 4)$, and evidence sentences $s^i_j, 0 \leq j \leq 6$ for a_i. I.e. inputs of each encoders is a evidence set $es_i =< q \circ a_i \circ s^i_1 \circ ... \circ s^i_6$>, where "$\circ$" is the concatenation operation.

All sentences are encoded to a hidden embedding.

$$h^{s_i} = h^{c_1} \circ h^{c_2} \circ ... \circ h^{c_{|s_i|}} \tag{2}$$

Therefore, the output of the encoders are the encoding of the entire text.

$$h^{es_i} = h^q \circ h^{a_i} \circ h^{s^i_1} \circ ... \circ h^{s^i_6} \tag{3}$$

$$h^{<D,Q,A>} = h^{es_1} \circ h^{es_2} \circ ... \circ h^{es_4} \tag{4}$$

For each kernel k in the kernel set K, EVERGREEN convolves the reshaped $h^{<D,Q,A>}$.

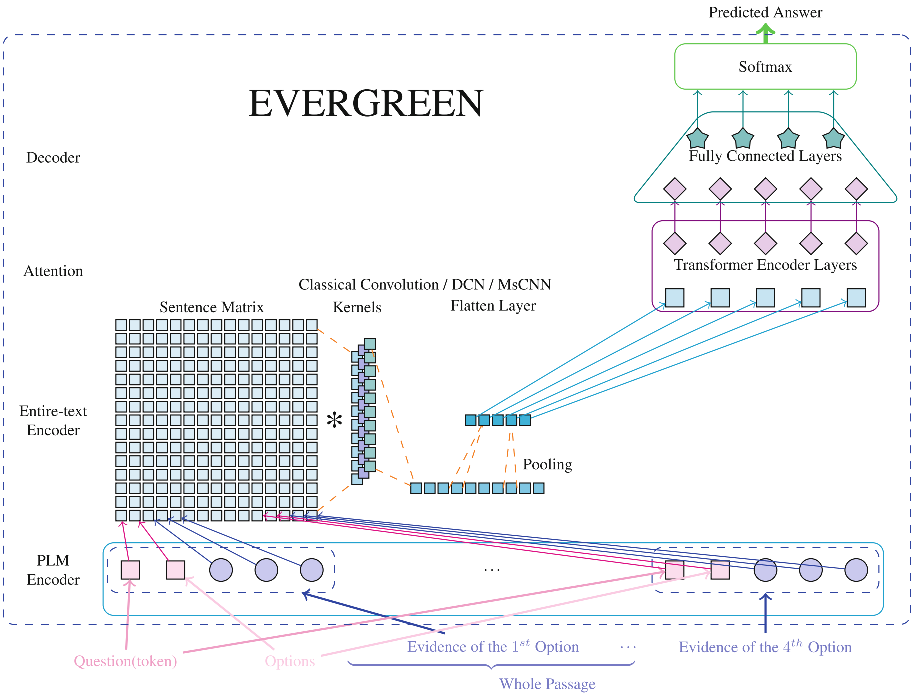


Fig. 4. The Architect of EVERGREEN.

$$h^k_{Conv} = Conv(reshape(h^{<D,Q,A>}), k) \tag{5}$$

All convolutions should be finished with a ReLU pooling layer.

$$H^k_{Conv} = \phi(h^k), where\, \phi(x) = max(0, x) \tag{6}$$

Results of the convolution with different kernels are concatenated.

$$H^{<D,Q,A>}_{Conv} = \Sigma_{k \in K} H^k_{Conv} \tag{7}$$

Then, this entire-text encoding will be fed to the standard attention mechanism.

$$ATT_\theta(Q_{att}, K_{att}, V) = softmax(\frac{Q_{att}K^T_{att}}{\sqrt{d_{k_{att}}}})V \tag{8}$$

where Q_{att} is a query vector, K_{att} is a key vector,V is a value vector, and $\sqrt{d_{k_{att}}}$ is the scale factor. The bilinear attention function ATT in Eq. (8) is used with parameters $\theta \propto \exp\left(WH^{<D,Q,A>}_{Conv} + B\right)$, where the weight is W and the bias is B.

At last, a fully connected network decoder (FCN) with Softmax is used to ensemble all branches' predictions while obtaining each option's score at the final layer, in which the loss function is the negative log-likelihood (NLL) of predicted answers, where $\hat{L} = l_1, l_2, l_3, l_4$ is the prediction, and L is the golden answer.

$$\hat{L} = softmax(FCN(ATT_{\theta}(H_{Conv}^{<D,Q,A>})) \tag{9}$$

$$LOSS_{NLL}(< D, Q, A >, L) = -\sum_{i=1}^{4} l_i \log P(l_i \mid D, Q, A) \tag{10}$$

Algorithm 1: Evidence Sets Building and Sentences Distribution Algorithm

```
input  : question token $q$,
         options $a_1, a_2, a_3, a_4$,
         passage $D = s_1, ...s_{|D|}$
output: evidence sets $es_1, es_2, es_3, es_4$
Initialize $es_i \leftarrow \emptyset, 0 < i \leq 4$;
for $i \leftarrow 1$ to $4$ do
    for $j \leftarrow 1$ to $|D|$ do
        Initialize $s_j$ as unused;
        Calculate sentence similarity $Similarity(a_i, s_j)$;
for $i \leftarrow 1$ to $4$ do
    $es_i \leftarrow q + a_i$;
    for $j \leftarrow 1$ to $|D|$ do
        if $Similarity(a_i, s_j)$ is the biggest (top evidence) then
            $es_i \leftarrow es_i \cup s_{j-1} \cup s_j \cup s_{j+1}$;
            mark $s_{j-1}, s_j, s_{j+1}$ as used;
    while $\Sigma_{s_k \in es_i} |s_k| < 512$ do
        if $s$ is unused and $Similarity(es_i, s)$ is the smallest then
            $es_i \leftarrow es_i \cup s$;
            mark $s$ as used;
return $es_1, es_2, es_3, es_4$
```

4.4 Evidence Extraction

EVERGREEN extract evidence sentences based on the cosine-similarity between the option and sentences in the passage. We try two evidence extraction strategies: first (Algorithm 1), locate the top relevant sentence for an option in the passage and extract it along with the sentence right before it and the sentence right next to it, and then fill in the left space with sentences which are the less similar with the already selected sentences; second, select the top relevant sentences for an option in the passage as evidence. The reason for only extract six sentences for an option is the space limitation. In Algorithm 1, we put the top evidence with its pre and next sentence into a current evidence set at line 6–11, right after an initialization process at line 1–5, and distribute sentences which are dissimilar with the current evidence set at line 12–15.

4.5 Entire-Text Convolution

EVERGREEN does the entire-text convolution. However, to verify the effectiveness of convolution, the component of EVERGREEN also supports MsCNN and DCN.

5 Experiments

5.1 Experiment Settings

The platform employed in the experiments is PyTorch 1.9.0 with Python 3.8.13 on Ubuntu 20.04.1 LTS, which exerts an Intel Core i7-17700 CPU with two RTX 3090 GPUs. Table 3 list all hyper-parameters of different models used in our experiments.

Table 3. Hyper-parameters of Models.

	EVERGREEN	BERT	Lonformer	T5
train batch size	4	4	4	4
dev batch size	4	4	4	4
test batch size	4	4	4	4
epoch	3	3	3	3
learning rate	2e-6	2e-6	2e-6	2e-6
gradient accumulation steps of SGD	1	1	1	1
seed	42	42	42	42

The dataset is divided into the training set, validation set, and test set according to the ratio 8:1:1. From both the length and question type aspect, the data distribution is consistent. I.e., not only the train, dev, and test sets keep similar passage and option length distribution, but also the distribution of select-true or select-false questions is almost identical. We further ensure the distribution of answers in these sets, see Table 4.

5.2 Model Comparison

Table 5 compares models which include BERT, AnchiBERT, GuwenBERT, Longformer, T5, and EVERGREEN. Inputs of these baselines are the question and options, along with a truncated passage (if necessary). Because most items in ACRE are shorter than 1536, we take the ensemble model of three PLMs to discover behaviors

Table 4. Answer Distribution in Training, Validation and Test Set.

Answer	Split			
	training	validation	test	total
A	822	102	102	1026
B	822	102	102	1026
C	821	102	102	1025
D	821	101	101	1023

of baselines. We used a stacking mechanism that facilitates a four-level fully connected network with input *logit* vectors from PLMs models. Each ensemble leverages three branches equipped with the same baseline model, i.e., three BERT models (tri-BERT), or three AnchiBERT models (tri-AnchiBERT), three GuwenBERT models (tri-GuwenBERT), or three MacBERT models (tri-MacBERT).

Table 5. Model Comparison on ACRE.

Mode Model	**original**[a]	**t. passage**[b]	**t. question**[c]	**f.w.e.**[d]
tri-BERT[e]	26.29	23.59	24.82	29.24↑
tri-AnchiBERT	28.75	27.76	27.03	25.80
tri-GuwenBERT	26.78	23.34	26.29	21.87
tri-MacBERT[f]	24.08	29.24↑	25.80↑	28.26↑
Longformer	23.83	25.06↑	26.54↑	28.75↑
T5	24.82	23.83	25.06↑	23.34
EVERGREEN-BERT	35.38	34.40	28.50	31.94
EVERGREEN-AnchiBERT	35.14	30.96	25.55	31.20
EVERGREEN-GuwenBERT	23.10	27.27↑	25.80↑	24.82↑
EVERGREEN-MacBERT	34.89	34.89	25.06	36.36↑
Human	32.00	N/A	N/A	N/A

[a] Original passages, question, and options.
[b] Passages are translated into modern Chinese.
[c] Passages with translated questions & options.
[d] Passages after function word erasing (f.w.e.).
[e] BERT-Base for Chinese.
[f] MacBERT(large).

We asked 184 10^{th}-grade students as users to take tests that are comprised random questions from ACRE. Every randomly sampled question is assigned to four students, and the average accuracy is 32%.

5.3 Results Analysis

Table 5 brings three observations:

1. translating AC passage into modern Chinese, make MacBERT, Longformer and EVERGREEN with GuwenBERT better, which indicates the advantage of translation on low-resource language datasets;
2. by contrast, translating modern Chinese options into AC is good for Longformer and T5, but can only slightly boost EVERGREEN with GuwenBERT as the base encoder;
3. removing function words from passages can improve the accuracy of many cases, such as using MacBERT on ACRE;

5.4 Ablation Test

Table 6 shows the results of ablation test of EVERGREEN on ACRE.

Table 6. Ablation Test on EVERGREEN (f.w.e.). Base model is a tri-MacBERT without evidence extraction or any convolution layer. Top^{1++} indicates the Top-1 evidence with its pre and next with sentences dissimilar to existing sentences.

Mode	Convolution	Accuracy %
base		28.26
base + Top^n evidence	Classical Convolution	34.40
	DCN	24.08
	MsCNN	22.11
base + Top^{1++}	Classical Convolution	36.36
	DCN	26.04
	MsCNN	22.11

Summarizing Table 6, we can draw the conclusion that classical convolution is the best and our sophisticated evidence extraction can slightly boost the accuracy.

6 Conclusion

This paper built the first dataset for the low-resource ancient Chinese reading comprehension task, ACRE, and proposed EVERGREEN, a PLM-based long-text-encoding-via-convolution model. The questions in ACRE are entire-text-regarding exception questions which distinguish intelligent people from others and are highly comprehensive. Experiments showed that ACRE is challenging, yet our newly proposed evidence extraction with sentence distribution approach can slightly boost the accuracy of EVERGREEN.

Two limitations of this paper are the limited size of ACRE and the to-be-improved accuracy of the proposed model. With all the variants we tried, the accuracy of EVERGREEN is still to-be-improved. It shows the difficulty of ACRE, but we believe there are better models. Therefore, besides keeping building ACRE, at least three attempts are on our schedule. First, elaborated sentences or span extraction algorithms might be helpful. Second, incorporating ancient Chinese knowledge [11] is another promising future direction. Third, we need a larger scale human evaluation.

Acknowledgements. This paper is supported by Guangdong Basic and Applied Basic Research Foundation, China (Grant No. 2021A1515012556).

References

1. Bartolo, M., Roberts, A., Welbl, J., Riedel, S., Stenetorp, P.: Beat the AI: investigating adversarial human annotation for reading comprehension. Trans. Assoc. Comput. Linguist. **8**, 662–678 (2020)
2. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
3. Dzendzik, D., Foster, J., Vogel, C.: English machine reading comprehension datasets: a survey. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8784–8804 (2021)
4. Jiang, Y., et al.: Improving machine reading comprehension with single-choice decision and transfer learning. arXiv abs/2011.03292 (2020)
5. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: Race: large-scale reading comprehension dataset from examinations. In: EMNLP (2017)
6. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: a lite bert for self-supervised learning of language representations. arXiv abs/1909.11942 (2020)
7. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
8. Putri, R.A., Oh, A.H.: IDK-MRC: unanswerable questions for Indonesian machine reading comprehension. In: The 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022. EMNLP (2022)
9. 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)
10. Sugawara, S., Inui, K., Sekine, S., Aizawa, A.: What makes reading comprehension questions easier? In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4208–4219 (2018)
11. Sun, K., Yu, D., Yu, D., Cardie, C.: Investigating prior knowledge for challenging Chinese machine reading comprehension. Trans. Assoc. Comput. Linguist. **8**, 141–155 (2020)
12. Tan, H., et al.: GCRC: a new challenging MRC dataset from Gaokao Chinese for explainable evaluation. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1319–1330 (2021)
13. Tian, H., Yang, K., Liu, D., Lv, J.: Anchibert: a pre-trained model for ancient Chinese language understanding and generation. In: Proceedings of the International Joint Conference on Neural Networks (2021)
14. Wang, S., Huang, M., Deng, Z., et al.: Densely connected CNN with multi-scale feature attention for text classification. In: IJCAI, vol. 18, pp. 4468–4474 (2018)

15. Xu, S., Liu, Y., Yi, X., Zhou, S., Li, H., Wu, Y.: Native Chinese reader: a dataset towards native-level Chinese machine reading comprehension. In: Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2022)
16. Yu, W., Jiang, Z., Dong, Y., Feng, J.: Reclor: a reading comprehension dataset requiring logical reasoning. In: International Conference on Learning Representations (ICLR) (2020)
17. Zeng, W., et al.: Pangu-α: large-scale autoregressive pretrained Chinese language models with auto-parallel computation. arXiv preprint arXiv:2104.12369 (2021)
18. Zhang, C., Lai, Y., Feng, Y., Zhao, D.: Extract, integrate, compete: towards verification style reading comprehension. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 2976–2986 (2021)
19. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable ConvNets V2: more deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)

Chinese Macro Discourse Parsing on Generative Fusion and Distant Supervision

Longwang He[1], Feng Jiang[2,3], Xiaoyi Bao[1], Yaxin Fan[1], Peifeng Li[1], and Xiaomin Chu[1](✉)

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
{lwhe,xybao,yxfan}@stu.suda.edu.cn, {pfli,xmchu}@suda.edu.cn
[2] School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
jeffreyjiang@cuhk.edu.cn
[3] School of Information Science and Technology, University of Science and Technology of China, Hefei, China

Abstract. Most previous studies on discourse parsing have utilized discriminative models to construct tree structures. However, these models tend to overlook the global perspective of the tree structure as a whole during the step-by-step top-down or bottom-up parsing process. To address this issue, we propose DP-GF, a macro Discourse Parser based on Generative Fusion, which considers discourse parsing from both process-oriented and result-oriented perspectives. Additionally, due to the small size of existing corpora and the difficulty in annotating macro discourse structures, DP-GF addresses the small-sample problems by proposing a distant supervision training method that transforms a relatively large-scale topic structure corpus into a high-quality silver-standard discourse structure corpus. Our experimental results on MCDTB 2.0 demonstrate that our proposed model outperforms the state-of-the-art baselines on discourse tree construction.

Keywords: Macro discourse analysis · Distant supervision · Generative fusion

1 Introduction

Discourse analysis is one of the fundamental tasks in natural language processing. A discourse is a linguistic entity composed of consecutive paragraphs or sentences, expressing a complete linguistic information. Discourse analysis is mainly divided into two levels: micro and macro. Micro-level discourse analysis focuses on the organizational structure and semantic relationship between sentences, while Macro-level discourse analysis examines the organizational structure and semantic relationship between paragraphs. Taking article chtb_0236 from the Macro Chinese Discourse Treebank (MCDTB) [1] as an example, the

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 159–171, 2024.
https://doi.org/10.1007/978-981-99-7022-3_15

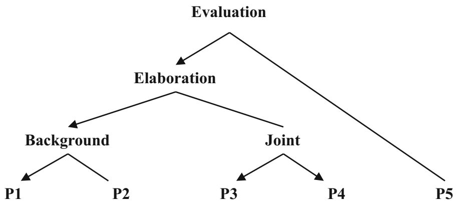


Fig. 1. Macro Discourse Structure Tree of chtb_0236

macro discourse structure is shown in Fig. 1. In the figure, the leaf nodes represent paragraph-level elementary discourse units (PDUs), while non-leaf nodes indicate the relationship between two adjacent discourse units (DUs).

The existing methods for discourse structure parsing mainly focus on three main aspects: enhancing the semantic representation of discourse units [6,9], strengthening the interaction between semantics [2,4], and improving construction methods [3,5,7,8]. However, all of these approaches view discourse structure parsing as a process-oriented task that requires incremental parsing of the structure to ultimately obtain a discourse structure tree. This process-oriented parsing method relies on the calculation of local semantic similarity and can easily neglect the overall understanding of the tree structure from a global perspective.

When parsing a document, the importance of comprehending the document as a whole cannot be ignored. Annotators must understand the theme and content of the document after reading it, to better grasp the structure and language characteristics of the document. Only with a global understanding can annotators accurately parse the document into a tree structure and convert it into a linear sequence as the learning objective of the model. Process-oriented parsing methods mainly simulate the human annotation process, while ignoring the global understanding stage. Therefore, it has become a challenge to explore how to use linear sequences representing tree structures to construct result-oriented parsing methods, and how to combine the advantages of process-oriented and result-oriented parsing methods.

In addition, due to the coarser granularity of macro-level discourse text and the more complex information it contains, the annotation process is very difficult, leading to a smaller corpus size. From the perspective of data, it cannot support the model in fully understanding the discourse information. Therefore, recent research has shifted to unsupervised and semi-supervised learning [10–13]. However, due to the limited supervision signal strength of unsupervised learning and semi-supervised learning, the performance cannot match that of supervised learning. Therefore, acquiring high-quality and large-scale datasets has become the second challenge.

We propose a Discourse Parser on Generative Fusion (DP-GF), which integrates result-oriented generative methods into traditional process-oriented methods, and combines the two methods for jointly learning while sharing the encoding layer. This method not only retains the advantages of process-oriented methods, but also models the entire article from a holistic perspective, avoiding the tedious tree-building process and more intuitively reflecting the structural features.

To address the second challenge, we propose a Distant Supervised Pretraining Method that transforms a relatively large topic-structured corpus into a silver-standard discourse structure corpus. This conversion method uses golden topic boundary information to ensure the quality of the converted discourse structure. The silver standard discourse structure corpus was used for pretraining the model, and the gold standard discourse structure corpus was used for incrementally training, greatly alleviating the small sample problem.

2 Related Work

Previous research on discourse structure parsing was mainly categorized as three frameworks: top-down, bottom-up, and bidirectional parsing frameworks.

For the first category, most of the relevant research is based on pointer network frameworks and proposes some effective strategies to enhance the representations of discourse units and semantic interaction between discourse units. In these studies, Lin et al. [4] first used a pointer network framework. Fan et al. [2] further proposed a pointer network that integrates global and local information to enhance semantic interaction. Koto et al. [5] defined the task as a sequence annotation problem, thereby eliminating the decoder and reducing the search space for segmentation points. Zhang et al. [14] regarded text parsing as a recursive split point sorting task and effectively improved the efficiency of the split point sorting task by encoding the split points in the pointer network. Zhang et al. [15] introduced a new method to convert the gold standard and prediction tree into a tree graph with two color channels.

For the second type, mainstream bottom-up frameworks are all transition-based methods. Mabona et al. [16] proposed a model based on beam search algorithm that can track structure and word generation actions. Zhou et al. [6] used the method of Shift-Reduce to extract macro discourse semantic information and construct a Chinese macro discourse structure tree from multiple views. Jiang et al. [7] utilized the left-branch bias characteristic of Chinese discourse structure to propose global and local reverse reading strategies to construct a discourse structure tree. Jiang et al. [8] explored a new construction method by introducing topic segmentation models into transition-based construction methods, improving the parsing capabilities of long texts.

The third framework combines the advantages of the first two frameworks, and there is relatively little research. Recently, He et al. [17] proposed a bidirectional parsing method that includes decision-makers, which can freely switch between splitting and merging actions, and select appropriate parsing actions. The above three frameworks are all process-oriented parsing methods, with a main focus on each step of parsing.

3 Discourse Parsing on Generative Fusion

To model discourse from both the process-oriented and result-oriented perspectives, we propose a Discourse Parser based on Generative Fusion (DP-GF). As shown in Fig. 2, the parser can not only be used as a discriminant model to perform bidirectional parsing using pointer networks but also can be used as a generative model to directly output tree structured linear sequences according to the original input. Due to the encoder-decoder architecture of T5 [18] being suitable for generating fusion methods [23], we have chosen it as the backbone of the DP-GF model. DP-GF mainly includes three parts: an encoder based on T5, a decoder based on the discriminant model, and a decoder based on the generative model. For an article, DP-GF generates two bare tree results and then uses the nuclearity and relationship classifier proposed by Lin et al. [4] for nuclearity and relationship recognition based on the tree structure.

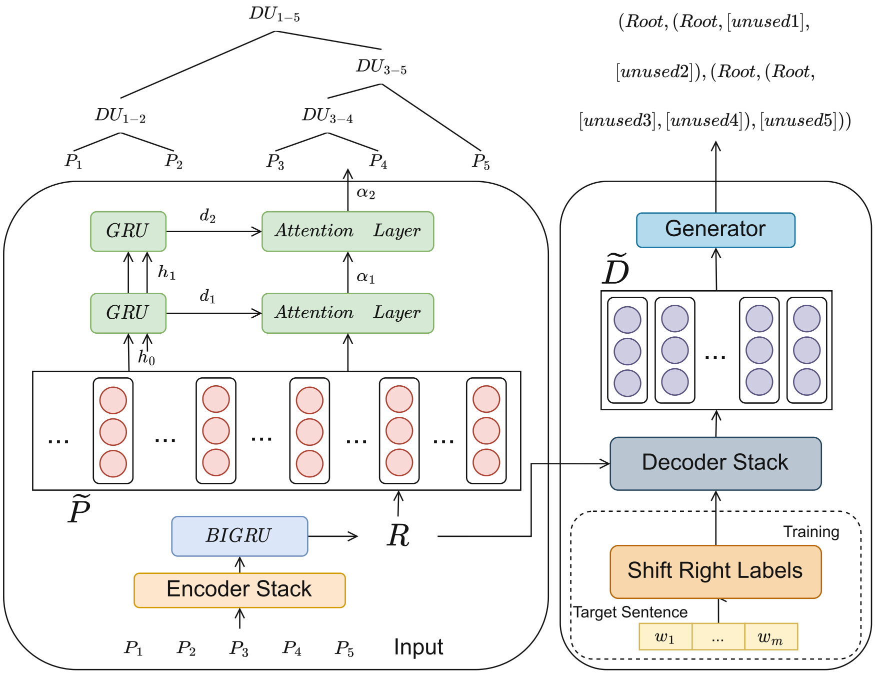


Fig. 2. The architecture of DP-GF model

3.1 T5-Based Encoder

Due to the shared encoder between two decoders, it means that DP-GF only needs to process one input information. We define the input part of DP-GF.

Given a document $T = \{t_1, t_2, \ldots, t_n\}$, where n is the number of PDUs in the document, and $t_s (1 \leq s \leq n)$ is the text of the s-th PDU.

We insert $[unusedx]$ at the beginning of each PDU, where $x(1 \leq x \leq n)$ is the PDU serial number, and then get $\widetilde{T} = \big\{[unused1], t_1, [unused2], \ldots [unusedn], t_n\big\}$. We enter $\widetilde{T}$ into $T5EncoderStack$[1] to obtain $R = \{r_1, r_2, \ldots, r_m\}$, where m is the total number of all words. Then we input R into BIGRU to obtain the final representation $P = \{p_1, p_2, \ldots, p_m\}$, and finally extract the representations of all $[unusedx]$ (x refers to 1 to m) positions to obtain the representation $\widetilde{P} = \{\widetilde{p}_1, \widetilde{p}_2, \ldots, \widetilde{p}_m\}$ of all PDUs.

3.2 Decoder on Discriminant Model

We first introduce the model architecture of UnifiedParser [4], which is the base model of the decoder. UnifiedParser adopted a pointer network parsing framework which is a typical process-oriented parsing method that recursively segments the span to obtain a discourse structure tree.

At each decoding step, the decoder takes the last DU representation in the span to be parsed and the hidden state h_{t-1} from the previous step as input to the Gated Recurrent Unit (GRU), obtaining the current decoder state d_t and hidden state h_t. h_t contains both the document-level representation of the full text as well as all the decoding information from the previous decoding step. The attention score is calculated based on d_t and the representation of the current span to be parsed $\tilde{P}_{STP}$, as follows.

$$\alpha_t = softmax(\sigma(d_t, \tilde{P}_{STP})) \tag{1}$$

where $\widetilde{P}_{STP}$ is the set of all position representations in the span to be parsed, and σ is the dot product operation. α_t represents the semantic connection closeness scores between adjacent DUs in STP. The higher the probability, the looser the semantic connection between DUs located beside the split position becomes. As a result, based on the probability distributions α_t of the output of decoder, we can obtain the split positions in step t.

3.3 Decoder on Generative Model

During training, the decoder based on the generative model has two inputs: the output R of the T5 encoder and the target sentence G, and outputs the decoder state $\widetilde{D}$. Since R has been obtained through the encoder, in this section we first introduce the construction of the target sentence G and then introduce the constrained decoding strategy.

Target Sentence Construction. The representation method for the target sentence should to be able to summarize the hierarchical structure and shape of the tree from a holistic perspective. We have drawn inspiration from the

[1] https://github.com/renmada/t5-pegasus-pytorch.

method of structured label embedding and made some modifications. Two types of vocabulary are selected to construct the target sentence, namely the *Root* Node representing the root node of each subtree and the $[unusedx]$ symbolizing the entire PDU representation at the beginning of each PDU in the input text.

Taking the tree structure in the upper left corner of Fig. 2 as an example, we first construct the bottom-level subtrees, namely DU_{1-2} and DU_{3-4}. Using $[unusedx]$ to represent Paragraph P_x, we then insert the *Root* Node in front of P_1 and P_2, as well as in front of P_3 and P_4, and then add parentheses to the outermost layer. This method is recursively used to merge subtrees and generate the linear sequence shown in the upper right corner of Fig. 2.

If the root node and bracket information are directly inserted between PDUs, the target sentence will be too long and truncated, resulting in information loss. However, using the $[unusedx]$ approach not only originates from the initial input text and is considered as a compressed representation of the entire paragraph, but also simplifies the generation difficulty of T5 model in subsequent tasks.

Constrained Decoding. Because the linear sequence generated during the inference process must comply with the convention, otherwise it cannot be transformed into a legitimate tree. In this study, we borrowed a tire-based constraint decoding algorithm [19,20] to achieve controllable text generation. Specifically, there are three candidate vocabulary items for each step: Parentheses, PDU serial number markers, and root nodes.

We maintain a stack and two counters, where the stack is used to store subtree states, and the two counters are used to store the number of unused root nodes and the number of leaf nodes, respectively, to control the generation of parentheses and root node characters. The vocabulary generated in step i will mask out the invalid vocabulary set V_i in step i, which is used to protect the legality of the generated sequence. Joint learning by adding the losses of two decoders during training.

4 Discourse Parsing on Distant Supervision

There is a clear similarity between topic structure and discourse structure [8], and the corpus of topic structure is easier to obtain compared to that of discourse structure. In the topic structure corpus, there is no structure within or between topics. Simple rules cannot generate a tree structure based on standard topic boundaries, while simple topic segmentation models cannot utilize standard topic boundary information. Therefore, we propose a distant supervision framework that combines rules and models, which can be used to generate a large-scale, silver standard macro-discourse structure corpus, as shown in Fig. 3.

This distant supervision framework consists of four steps: 1) training a topic segmentation model using a topic structure corpus; 2) converting the topic structure corpus into a silver-standard discourse structure corpus using the topic segmentation model; 3) pre-training the discourse structure parsing model using the silver-standard discourse structure tree as the pre-training dataset; 4) fine-tuning

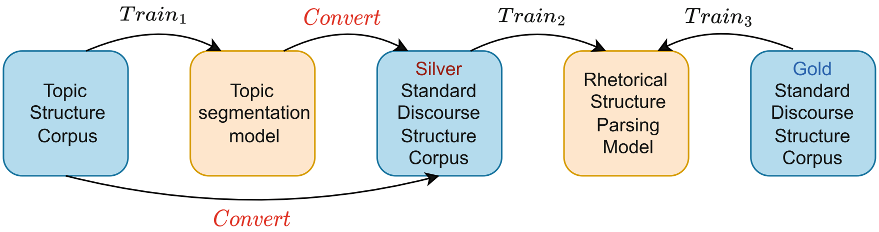

Train1
Convert
Train2
Train3
Topic Structure Corpus
Topic segmentation model
Silver Standard Discourse Structure Corpus
Rhetorical Structure Parsing Model
Gold Standard Discourse Structure Corpus
Convert

Fig. 3. Discourse parsing on distant supervision

the discourse structure parsing model on the gold-standard discourse structure corpus. The topic segmentation model uses the TM-BERT triple semantic matching model proposed by Jiang et al. [8]. The convert method of the silver-standard discourse structure corpus is shown in Fig. 4.

Relying solely on the gold topic boundaries is not enough to convert topic structure into discourse structure, as it is unable to establish connections between the subtrees. Therefore, a topic segmentation model is needed to further predict the segmentation probabilities between the subtrees. The topic segmentation model is used to perform a binary classification task, which is to judge whether there is a topic transition or continuation at the end of each paragraph. Therefore, the segmentation probabilities are predicted at the boundary positions of each paragraph. Meanwhile, the document also has gold-labeled topic boundaries, and we need to use these boundaries to divide the document into several subtrees, maximizing the quality of the discourse structure tree. Therefore, in the case of knowing the gold topic boundaries in advance, the gold segmentation probability at the segmentation point position is set to 1, and 0 is set at non-segmentation point positions. The two probabilities are added together to obtain the final probability. Finally, the segmentation points are sorted in Descending order according to the final probability, and the discourse structure tree is obtained by cutting at the sorted segmentation points.

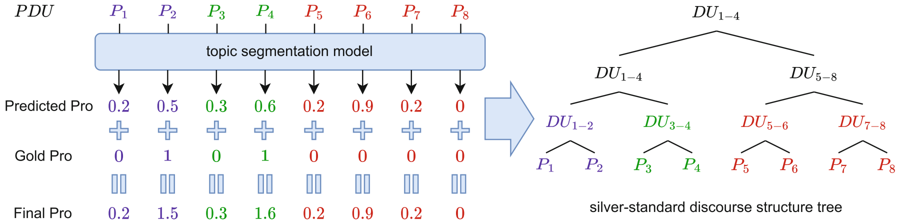


Fig. 4. The convert method of the silver-standard discourse structure corpus

Taking the document on the left side of Fig. 4 as an example, the document has 8 paragraphs. P_1 and P_2 belong to the same topic, P_3 and P_4 belong to the same topic, and the remaining four paragraphs belong to the same topic.

According to the above segmentation probability calculation rule, the gold topic boundary positions are always given priority for segmentation. Among the gold topic boundary positions, the positions with higher final probabilities have higher priorities. Therefore, the cutting is first performed after the fourth paragraph, dividing the eight paragraphs into two subtrees, the first four paragraphs and the last four paragraphs. This rule is recursively applied to the subtrees that have not been completely segmented, resulting in the discourse structure tree on the right side.

5 Experimentation

5.1 Dataset and Experimental Settings

To obtain an extraterritorial topic structure corpus, we collected 14393 Chinese news corpora from Xinhua News Agency from the Gigaword corpus[2]. Each document has several subheadings as explicit topic boundaries. By removing all subheadings, a topic structure corpus is obtained. Finally, a silver standard discourse structure corpus is constructed using the transformation method in Sect. 4.

The dataset and evaluation metrics used in this study are consistent with He et al. [17]. MCDTB 2.0 is an expanded version of MCDTB 1.0, and its annotation process is highly consistent with MCDTB 1.0. MCDTB 2.0 contains 1200 articles, with an average length longer than MCDTB, which further tests the model's generalization ability. We report micro-averaged F1 scores for predicting span attachments in discourse tree construction (Span), span attachments with nuclearity (Nuclearity), and span attachments with relation labels (Relation). Specifically, we evaluate the nuclearity with three classes, and we use 15 finer-grained types for evaluation in relation classification.

In the pre-training stage, the model architecture and parameters of the decoder based on the discriminative model used in this study are the same as those of UnifiedParser [4], with a learning rate of 1e-5. The learning rate of the generative decoder is 5e-4. The maximum input length is 512, and the model is trained for 40 epochs. In the fine-tuning stage, the learning rates of the two decoders are 1e-4 and 5e-3, respectively, and the model is trained for 10 epochs.

5.2 Experimental Results

We compare the proposed model with various strong baselines as follows.

- **UnifiedParser** [4]: a parser incorporating information from parent and sibling nodes.
- **GBLRR** [7]: a parser that inverts the order of parsing to achieve reverse reading.
- **MDParser-TS** [8]: a parser that uses the topic segmentation method.

[2] https://catalog.ldc.upenn.edu/LDC2009T2.

- **DGCNParser** [9]: a parser that used topic graphs to model the semantic relationships within and between DUs.
- **AdverParser** [15]: a SOTA model on micro-level, which converted predicted trees and gold trees into graphs and trains an adversarial bot to exploit global information.
- **Vanilla T5**: a baseline using the T5 framework is proposed in this paper, without using generative fusion methods and distant supervision methods.
- **UnifiedParser(T5)**: Similar to the UnifiedParser introduced above, the only difference is to replace the encoder with the encoder of T5.

Table 1. Performance comparison of discourse tree construction (Micro F1)

Model	Pre-training Models	Span	Nuclearity	Relation
UnifiedParser	ELMo	52.64	36.92	31.85
GBLRR	BERT	61.87	54.25	28.35
MDParser-TS	BERT	59.68	45.76	27.95
DGCNParser	BERT	61.98	49.95	28.97
AdverParser	XLNet	64.64	**57.96**	**40.26**
Vanilla T5	T5	59.16	46.49	27.48
UnifiedParser(T5)	T5	62.58	48.23	29.62
DP-GF$_{ours}^{gen}$	T5	66.19	53.97	36.89
DP-GF$_{ours}^{dis}$	T5	**69.02**	56.89	37.96

The results are shown in Table 1. $DP-GF_{ours}^{gen}$ and $DP-GF_{ours}^{dis}$ are our two models, which respectively represent the results generated by $DP-GF$ using the generative and discriminative methods with five-fold cross-validation. $DP-GF_{ours}^{dis}$outperformed all baselines in terms of structural performance, with a 4.38 improvement over the previous state-of-the-art model AdverParser, also significantly better than the two baseline models that used T5. The performance on nuclearity and relation are slightly lower than that of AdverParser, which is because AdverParser added the nuclearity and relation channel in the adversarial graph, while our method only focuses on the span. Despite having only half the input length of AdverParser, more truncated text, and greater loss of information, our method demonstrates better performance, demonstrating its effectiveness. One reason for this is that the T5 pre-training model has larger model parameters and corpus scale in the pre-training stage compared to XLNet, making it superior. In addition, distant supervision and joint learning also play an indispensable role. The performance of $DP-GF_{ours}^{gen}$ is slightly inferior in comparison, but as a preliminary attempt at a generative method, its performance remains comparable.

5.3 Ablation Analysis

In order to investigating the effectiveness of distant supervision and generation fusion framework, ablation experiments were conducted and analyzed in this section. As shown in Table 2, the first four rows represent the use of distant supervision, while the last four rows represent the removal of distant supervision. The first two rows in each group employ the joint framework of generation and discrimination, while the last two rows did not use the joint model of generation and discrimination.

Table 2. Ablation analysis

		Model	Span
w/ Distant Supervision	w/ Joint Model	DP-GF$_{gen}$	66.38
		DP-GF$_{dis}$	**69.04**
	w/o Joint Model	Vanilla T5	63.87
		UnifiedParser(T5)	65.94
w/o Distant Supervision	w/ Joint Model	DP-GF$_{gen}$	62.09
		DP-GF$_{dis}$	64.51
	w/o Joint Model	Vanilla T5	59.16
		UnifiedParser(T5)	62.58

It can be observed from the experimental results that both methods significantly improve the model performance, and distant supervision brought greater improvement. In addition, the fusion of the two methods is superior to using them separately. The effectiveness of the two methods has been verified, indirectly validating the quality of the silver-standard rhetorical structure corpus.

5.4 Analysis on Different Target Sentences

As generative models offer a high flexibility in constructing target sentences, we summarize some construction methods used in other fields. For example, the target sentence generation method based on GAS templates [21] primarily inserts special markers in the input sentence, replacing supervised signals with markers and positional information. The target sentence generation method based on Paraphrase templates [22] mainly generates action sequence processes. These two methods are both process-oriented methods for generation.

Figure 5 shows the application of these two templates in the structure parsing task. Taking the tree structure in the figure as an example, the target sentence generation method based on the GAS template primarily converts the text of each PDU into a [unused] tag and arranges them in order. Then, following the top-down splitting order of the structure tree, it inserts the sequence tag “Top” at each position. Meanwhile, the target sentence generation method based on

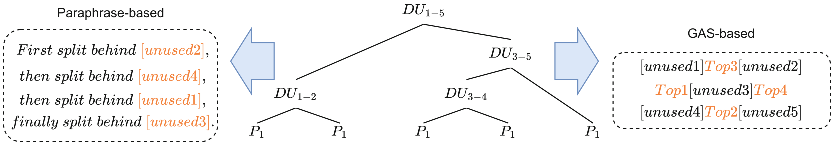


Fig. 5. Differences in different target sentence templates

Table 3. Differences in different target sentence templates.

Approach	Span
DP-GF$_{gen}$	66.38
DP-GF$_{dis}$	**69.02**
DP-GF$^{gen}_{GAS}$	62.98
DP-GF$^{dis}_{GAS}$	65.16
DP-GF$^{gen}_{Paraphrase}$	63.45
DP-GF$^{dis}_{Paraphrase}$	64.87

the Paraphrase template describes the main process of recursive splitting to construct the tree structure. These two methods are incorporated into DP-GF, and the constraint decoding method proposed in this chapter is added to ensure the legality of the generated sentences. The results obtained are shown in Table 3.

6 Conclusion

In this paper, we propose a macro-level discourse structure parsing method based on the distant supervision and generation fusion. This method can integrate high-quality data within the domain and large-scale data outside the domain, using both process-oriented and result-oriented approaches for discourse structure parsing. Meanwhile, the proposed constraint decoding algorithm can protect the legality of the generated sequence, resolving the problem of small-sample discourse structure parsing and the lack of a holistic parsing perspective. Experimental result shows that, compared to all baselines, our proposed model can effectively alleviate these two major problems. Our future work will focus on how to introduce more effective prompts to macro discourse parsing.

References

1. Jiang, F., Xu, S., Chu, X., et al.: MCDTB: a macro-level Chinese discourse treebank. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3493–3504 (2018)
2. Fan, Y., Jiang, F., Chu, X., et al.: Combining global and local information to recognize Chinese macro discourse structure. In: Proceedings of the 19th Chinese National Conference on Computational Linguistics, pp. 183–194 (2020)

3. Liu, L., Lin, X., Joty, S., et al.: Hierarchical pointer net parsing. arXiv preprint arXiv:1908.11571 (2019)
4. Lin, X., Joty, S., Jwalapuram, P., et al.: A unified linear-time framework for sentence-level discourse parsing. arXiv preprint arXiv:1905.05682 (2019)
5. Koto, F., Lau, J.H., Baldwin, T.: Top-down discourse parsing via sequence labelling. arXiv preprint arXiv:2102.02080 (2021)
6. Zhou, Y., Chu, X., Li, P., et al.: Constructing Chinese macro discourse tree via multiple views and word pair similarity. In: Natural Language Processing and Chinese Computing: 8th CCF International Conference, pp. 773–786 (2019)
7. Jiang, F., Chu, X., Li, P., et al.: Chinese paragraph-level discourse parsing with global backward and local reverse reading. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5749–5759 (2020)
8. Jiang, F., Fan, Y., Chu, X., et al.: Hierarchical macro discourse parsing based on topic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 13152–13160 (2021)
9. Fan, Y., Jiang, F., Chu, X., et al.: Chinese macro discourse parsing on dependency graph convolutional network. In: Natural Language Processing and Chinese Computing: 10th CCF International Conference, pp. 15–26 (2021)
10. Kobayashi, N., Hirao, T., Nakamura, K., et al.: Split or merge: which is better for unsupervised RST parsing? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 5797–5802 (2019)
11. Huber, P., Carenini, G.: Unsupervised learning of discourse structures using a tree autoencoder. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 13107–13115 (2021)
12. Nishida, N., Nakayama, H.: Unsupervised discourse constituency parsing using Viterbi EM. Trans. Assoc. Comput. Linguist. **8**, 215–230 (2020)
13. Kobayashi, N., Hirao, T., Kamigaito, H., et al.: Improving neural RST parsing model with silver agreement subtrees. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2021, pp. 1600–1612 (2021)
14. Zhang, L., Xing, Y., Kong, F., et al.: A top-down neural architecture towards text-level parsing of discourse rhetorical structure. arXiv preprint arXiv:2005.02680 (2020)
15. Zhang, L., Kong, F., Zhou, G.: Adversarial learning for discourse rhetorical structure parsing. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 3946–3957 (2021)
16. Mabona, A., Rimell, L., Clark, S., et al.: Neural generative rhetorical structure parsing. arXiv preprint arXiv:1909.11049 (2019)
17. He, L., Jiang, F., Bao, X., et al.: Bidirectional macro-level discourse parser based on oracle selection. In: PRICAI 2022: Trends in Artificial Intelligence: 19th Pacific Rim International Conference on Artificial Intelligence, pp. 224–239 (2022)
18. 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)
19. Chen, P., Bogoychev, N., Heafield, K., et al.: Parallel sentence mining by constrained decoding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1672–1678 (2020)
20. Lu, Y., Lin, H., Xu, J., et al.: Text2event: controllable sequence-to-structure generation for end-to-end event extraction. arXiv preprint arXiv:2106.09232 (2021)

21. Zhang, W., Li, X., Deng, Y., et al.: Towards generative aspect-based sentiment analysis. 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. 504–510 (2021)
22. Zhang, W., Deng, Y., Li, X., et al.: Aspect sentiment quad prediction as paraphrase generation. arXiv preprint arXiv:2110.00796 (2021)
23. Jiang, F., Fan, Y., Chu, X., et al.: Not just classification: recognizing implicit discourse relation on joint modeling of classification and generation. In: Proceedings of the. Conference on Empirical Methods in Natural Language Processing, pp. 2418–2431 (2021)

GHGA-Net: Global Heterogeneous Graph Attention Network for Chinese Short Text Classification

Meimei Li[1,2], Yuzhi Bao[1,2], Jiguo Liu[1,2](✉), Chao Liu[1,2], Nan Li[1,2], and Shihao Gao[1,2]

[1] Chinese Academy of Sciences, Institute of Information Engineering, Beijing, China
liujiguo@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. As an important research content in the field of natural language processing, Chinese short text classification task has been facing two challenges: (i) existing methods rely on Chinese word segmentation and have insufficient semantic understanding of short texts; (ii) there is lacking of annotated training data in practical applications. In this paper, we propose the Global Heterogeneous Graph Attention Network (GHGA-Net) for few-shot Chinese short text classification. First, we construct the global character and keyword graph representations from the entire original corpus to collect more text information and make full use of the unlabeled data. Then, the hierarchical graph attention network is used to learn the contribution of different graph nodes and reduce the noise interference. Finally, we concatenate embedding with text vector and fuse the keyword and character features to enrich the Chinese semantics. Our method is evaluated on the Chinese few-shot learning benchmark FewCLUE. Extensive experiments show that our method has achieved impressive results in the classification tasks of news text and sentiment analysis, especially in minimal sample learning. Compared with existing methods, our method has an average performance improvement of 5% and less training consumption, which provides a new idea for few-shot Chinese natural language processing without relying on pre-training.

Keywords: Chinese short text classification · Few-shot learning · Heterogeneous graph · Hierarchical graph attention · Feature integrate

1 Introduction

Short text classification (STC) is applied in many research scenarios, such as sentence pair matching [1], news classification [2] and sentiment analysis [3]. Different from the long text which includes several paragraphs, short text generally only contain one or a few sentences. Due to its length limitation, short

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 172–184, 2024.
https://doi.org/10.1007/978-981-99-7022-3_16

text cannot carry as rich semantic and grammatical information as long text. The fragmented text makes it difficult to obtain information beyond single word semantics, and it is almost impossible to understand text in combination with context. So STC task is much harder than long text when proper nouns appear in the text or some words have multiple meanings. Many studies based on graph neural network [4] aim to enrich the semantic information of short texts. The HGAT introduced [5] HIN structure builds graph based on the topic, entity and documents information and STGCN [6] uses words. However, topic acquisition and entity recognition methods cannot achieve high accuracy and requires additional training consumption. Others introduce part-of-speech (POS) tags [7] or use external wiki knowledge [8]. But these methods ignore the global text information in the original documents and have deviations in semantic understanding while Chinese texts carry more complex semantic information.

In natural language processing (NLP), the biggest difference between Chinese and English is that the character in English do not express meaning in most cases but Chinese did. For example, a text in TNEWS is "现实中的大司马是什么样的? (What is Da Sima like in reality?)", its category belongs to the game because "大司马(Da Sima)" is a game anchor. However, the "司马(sima)" was a type of official position in ancient China, and "马" is translated to horse directly. So that the complex meanings of Chinese words and characters are the biggest difficulty in Chinese STC and separate words from sentence in Chinese is much harder than English. The main way to solve this gap is to combine learning word and character features from Chinese text [2,9]. And there are also methods integrate sentences and words feature [10]. Lexicon [11] can match word through the tree structure more accurate, but it rely on external vocabulary.

General neural network methods [1] rely on large amount of training data to learn text features and perform poor while lacking labeled data. However, the cost of manually annotating all texts is unacceptable in practical STC tasks, while the extreme zero-shot learning rely heavily on pre-training and unable to adapt to multiple domains. In contrast, few-shot learning [12] only need a small amount of annotated texts and could achieve similar performance as normal.

To address the aforementioned problems, in this paper, we propose a **G**lobal **H**eterogeneous **G**raph **A**ttention Networks (GHGA-Net) for few-shot Chinese STC. By building the global heterogeneous graph, we make full use of the unlabeled texts information from entire original corpus to better fit few-shot learning. Then, we use the hierarchical graph attention networks to learn the contributions of different nodes to text categories and integrating word and character features to achieve deep understanding of the semantics of Chinese short texts.

The main contributions of this paper are summarized as follows :

- We propose the GHGA-Net method, which constructs heterogeneous graph to integrate keyword and character features to better represent the semantic information of Chinese short text. Graph attention mechanism is used to learn the contribution of different nodes and reduce noise interference.
- The unlabeled data are fully used by generating the global graph representation, which deeply collect the global semantic information of the original

data without pre-training and optimize the classification learning in the small number of annotation scenarios.
- The experimental results on the FewCLUE datasets show that the proposed method significantly improves the classification performance compared with other existing models.

2 Related Work

Graph Network for STC: In the text classification task, the global structure of the graph can model the complex semantic relationship between words in the text, and it is one of the most effective research methods to transform the text into a document graph and then use the graph neural network for learning and training. The graph convolutional neural network (GCN) [13] adds convolution operation to graph network, which can effectively compress the size of the model and increase the input size of text. The SHINE [7] model use GCN to combine documents, entities and position features, but it ignore the global information of short text. Addressing the lack of short text information, SimpleSTC introduce the external wiki text to enrich the global information, which benefits the STC task effectively. Attention mechanism is also applied to graph neural networks [14]. HyperGAT [15] introduces the concept of supergraph into text representation and uses dual attention mechanism to learn the nodes and edges of the graph respectively. Methods based on graph neural network can better represent the various feature information of short text. However, there is lacking in-depth research for Chinese STC based on graph neural network.

Pre-training for STC: In order to reduce training costs and adapt to more NLP tasks, pre-training models have been widely used in recent years [16–18]. These models are usually pretrained on large-scale corpora, enabling them to be more generalized and adaptable to few-shot learning scenarios. Thus, simply fine-tune the target dataset can achieve good results. However, most of pre-training models have large parameters and there are limitations in the actual deployment and operation process. Moreover, many models based on BERT has not made special optimizations in Chinese word segmentation, and it is still character segmentation, which hinders the understanding of Chinese semantics.

Chinese STC: Due to the particularity of Chinese text, the research based on integrate the word and character features of the text [9] has achieved good application results, and there are also methods to hierarchical learning sentence and words [10]. In addition, since the radicals of Chinese characters also belong to hieroglyphs, the radicals can also be added as a feature to the construction of Chinese text representation [2], but these methods are limited by embedding special word vectors. It is a valuable way to express Chinese text in the form of text map and integrate Chinese character and word features for learning.

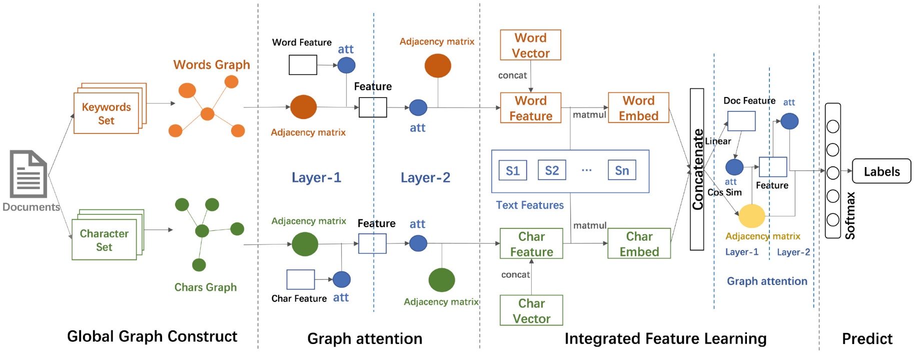


Fig. 1. The overall architecture of GHGA-Net Model.

3 Proposed Method

First, we give the task definition. For the given Chinese short text set $S_{doc} = (text)_n$ and its training set $S_{train} = (text, label)_m$, where $m << n$. Our goal is to train the classification model under the training set S_{train}, and finally predict the class label of remaining texts in S_{doc}.

The architecture of our GHGA-Net is shown in Fig. 1. Our idea is to extract the keywords and characters from each text in the whole corpus S_{doc} to construct the global graph representation. The hierarchical graph attention network is introduced to learn the graph features, which weighted the original graph representation and word embedding. Then our method fuse the heterogeneous features to document feature. Another hierarchical graph attention layer update the feature and the final prediction is made through softmax.

3.1 Global Graph Representation

In the case of only a small number of sample annotations, relying solely on training data to construct text features is clearly not enough. The unlabeled text in the original dataset can also be learned as implicit features to better obtain the semantic and category features of the text. So we choose to use the entire text set to construct the global graph representations.

Not all words contain specific information in Chinese. Therefore, we traverse each text in S_{doc}, extract and segment words of different parts of speech based on the term frequency-inverse document frequency (TF-IDF) and finally construct a global words vocabulary, only nouns, gerunds and some proper nouns under Chinese grammar are retained. Then, S_{doc} is cleaned according to the obtained global vocabulary. Next, we use point-wise mutual information (PMI) to calculate the word co-occurrence relationship between each keyword in vocabulary [13]. Let v_i, v_j be different keyword nodes in the global vocabulary, the relationship calculation method between them follows the following formula :

$$[C_{word}]_{ij} = max(PMI(v_i, v_j), 0) \tag{1}$$

C_{word} is a vector space with vocabulary length dimension in both rows and columns, which records the relationship between each node and other nodes in the global vocabulary. For each word in the global vocabulary, we match it with the pre-trained word2vec word vector to construct the word vector map.

According to the differences in grammar structure between Chinese and English. Besides word features, using character as the feature input of Chinese text classification can enrich the semantic and grammatical information of text. Therefore, we also propose and construct a global character vocabulary. For each short text in S_{doc}, we remove the numbers and symbols, only remain common words with word frequency above 10 and match with pre-trained character vectors. Similarly, the relationship $[C_{char}]_{ij}$ in character vocabulary is calculated by formula 1. Finally, we obtain the global features of keywords G_{gword} and characters G_{gchar} of documents with matched word vectors.

3.2 Hierarchical Graph Attention

In short text, not all words contribute same to the category information, especially in the case of lacking text information. To better focus on key features and reduce the interference of noise, we added the attention layers to update the weights of different nodes and perform weighted summation output.

For the constructed global heterogeneous graph representations C_{word} and C_{char}, the word vector V_{word} and V_{char}, we update the node vector H based on the two-layer graph attention networks:

$$H = GAT(C, ReLU(GAT(C, V))) \tag{2}$$

where RELU is the activation function, representing $[ReLU(x)]_i = max([x]_i, 0)$.

We directly introduce the pre-trained word vector here. Specifically, we regard the relation graph matrix as the input node vector, and the word vector embedding as the node feature. Performing a linear transformation on the node embedding $h_i^{(l)}$ in l-layer, similar to direct weighting in convolution operations [4], $W^{(l)}$ is a trainable weight parameter :

$$z_i^{(l)} = W^{(l)} h_i^{(l)} \tag{3}$$

Unlike concatenating the embedding of two nodes [14], our method uses a similar self-attention mechanism to calculate the original attention score for word nodes and character nodes respectively :

$$e_i^{(l)} = LeakyReLU(\vec{a}^{(l)T} z_i^{(l)}) \tag{4}$$

The attention weight is obtained by applying the softmax operation to the original attention score of the node. Finally, the features of all adjacent nodes are weighted and summed based on the attention weight:

$$a_i^{(l)} = \frac{exp(e_i^{(l)})}{\sum_{k \in N(i)} exp(e_k^{(l)})} \tag{5}$$

$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} a_i^{(l)} z_i^{(l)}) \tag{6}$$

For the weighted word encoding, we concatenate it with the word vector again to make better use of the semantic information between words. Finally, the graph representation is transformed into character embedding and keyword embedding.

$$E = concat(E, E_{bedding}) \tag{7}$$

3.3 Integrated Heterogeneous Feature Learning

At last, we learn the text features based on global heterogeneous text graphs. For each content of the original Chinese short text, the text graph G_{text} is constructed by transforming the text into a vector. The word features of the text are encoded as aggregation nodes and embedded into h_{wi}. The text relationship after word segmentation is calculated by:

$$h_{wi} = \beth(E^T s_i), [s_i]_m = TF - IDF(v_m, x_i) \tag{8}$$

where T stands for matrix transpose operation, $\beth$ stands for regularized $x/||x||_2$, The TF-IDF vector is calculated by [19], where v_m represent the nodes in G_{gword} and x_i represent the nodes in G_{text}. Words in text but not belong to the global vocabulary will not be calculated. Note that the character feature encoding h_c is also calculated using the same way. The final fusion text representation is concatenate encoded for word embed and character embed:

$$h = concat(h_w, h_c) \tag{9}$$

Our original intention is to use a similar way to graph attention network, where word and character embedding are input as adjacent nodes, and an additional attention layer is added to achieve feature fusion. However, due to the difference between Chinese words and characters, the attention method did not lead in all test datasets, while the concatenate operation generally achieved good results. The specific ablation study will be discussed in Sect. 4.4.

After obtaining the fused text coding, we first use linear transformation to obtain the feature vector F of the text, and then calculate the adjacency matrix A of the text based on cosine similarity :

$$F = linear(h) \tag{10}$$

$$[A]_{ij} = ReLU(cos(h_i, h_j) - \tau) \tag{11}$$

τ is the correlation threshold, and the final text category prediction is also updated by the two-layer GAT method we proposed:

$$Prediction = SoftMax(GAT(A, ReLU(GAT(A, F))) \tag{12}$$

SoftMax represents $[softmax(x)]_i = exp([x]_i)/\sum_j exp([x]_j)$. Finally, we use the cross entropy as loss function for optimization process of the model.

$$Loss = -\sum_{i\in\iota_l}(y_i)^T log(y_i) \tag{13}$$

The complete procedure of GHGA-Net is described in Algorithm 1:

Algorithm 1: GHGA-Net Algorithm

```
Input: short text dataset $S_{doc}$, global graph set $G$, pretrained embedding $E_{pre}$
Output: predict label list $L = l_1, l_2, ..., l_n$ and trained model
for $G$=$G_{gword}$, $G_{gchar}$ do
    update and generate the word embedding $E_w$ and character embedding $E_c$
      by (2)
    for $E$=$E_w$, $E_c$ do
        concatenate with the pretraind embedding by (7)
    end
end
for $E$=$E_{word}$,$E_{char}$ do
    obtain the aggregated heterogeneous text graph feature by (8)
end
fuse the word and char embedding to final text embedding $h$ by (9)
generate the text feature $F$ and adjacency matrix $[A]_{ij}$ by (10), (11)
update learning final text representation and predict the label by (12)
optimize model parameter by (13)
```

4 Experiments

4.1 Datasets

We conducted experiments on short text classification datasets from the Chinese few-shot learning benchmark FewCLUE [12] (Table 1):

1. **TNEWS**: The headline Chinese news short text classification dataset for few-shot learning tasks contains a total of 15 categories.
2. **EPRSTMT**: E-commerce product sentiment analysis dataset for sentiment polarity binary classification.

Table 1. Summary of used FewClue datasets.

Dataset	TrainSingle	TrainAll	DevAll	Classes	Unlabeled	LenAvg
EPRSTMT	32	160	160	2	20000	22
TNEWS	240	1185	1098	15	19565	36

4.2 Experimental Setup

BaseLines. We compare our method with the following three kinds of baselines:

- **General Method**: (1) **TextCNN**: Sentence classification method based on convolutional neural network [1]. (2) **BiLSTM-Att**: Bidirectional long short-term memory network with attention mechanism [20]. (3) **Transformer**: Encoder-decoder structure with multi-head attention [21].
- **Pre-training Model**: (1) **BERT**(Bert-wwm-Chinese): Pre-training model based on bidirectional Transformer architecture [16]. (2) **BERT-CNN**: Text encode by BERT and use CNN to train. (3) **RoBERTa**(RoBerta-wwm-Chinese): A robustly optimized BERT pre-training approach [18]. (4) **ERNIE**: Baidu's Pre-training model for Chinese natural language processing [17].
- **Graph Based Method**: (1) **HyperGAT**: Hypergraph attention neural network classification method based on LDA algorithm to extract text topics [15]. (2) **SimpleSTC**: GCN based short text classification method with external wiki knowledge [8].

4.3 Performance Comparison

Table 2 shows the performance. It can be seen that TNEWS is harder to classify due to its larger amount of categories. Our GHGA-Net achieves optimal results in almost all tasks and reaches an average improvement of about 5% compared with the second best baseline. Original methods achieve the worst average performance. All pre-training models perform well, and the RoBERTa model has achieved the highest accuracy on TrainAll set in TNEWS, which proves the advantages of using a large amount of corpus for pre-training in few-shot Chinese STC tasks. For graph based methods, HyperGAT performs obviously worse under small samples while SimpleSTC improves a little by external wiki knowledge. Besides, both of them are unable to deeply understand the complex semantics contained in Chinese. Our GHGA-Net is optimized for the semantic features of Chinese text, which integrates the heterogeneous graph features and introduce the hierarchical graph attention, receives the best result.

In the case of minimal training samples (TrainSingle), our method achieves state-of-the-art in both news multi-classification and sentiment binary classification tasks, which outperforms the second best baseline model by 6%. Almost all non pre-trained methods have a significant reduction in accuracy under extremely few-shot learning, which indicates their strong dependence on training

Table 2. Test performance (%) mesured on FewCLUE datasets. Normally trained under TrainAll data, the * mark represant trained in TrainSingle data. The best results are marked in bold, and the second-best results are underlined. The last row records the relative improvement of GHGA-Net over best results among other methods.

	TNEWS		EPRSTMT		TNEWS*		EPRSTMT*	
Model	ACC	F1	ACC	F1	ACC	F1	ACC	F1
TextCNN	44.08	44.36	50.78	50.78	23.59	21.18	49.38	47.89
BILSTM-ATT	14.30	9.18	41.41	40.97	7.74	3.28	45.62	45.52
Transformer	14.31	13.11	53.12	50.77	6.83	1.01	53.75	53.68
BERT	51.09	49.67	50.00	44.59	44.54	43.56	49.38	44.65
BERT-CNN	46.08	44.90	51.56	34.02	44.46	44.03	53.75	34.69
RoBERTa	**52.55**	<u>51.16</u>	49.22	45.12	<u>45.26</u>	<u>44.69</u>	48.13	44.85
ERNIE	<u>51.73</u>	51.13	47.66	34.71	42.17	40.22	46.88	35.58
HyperGAT	33.70	32.99	<u>65.62</u>	<u>65.46</u>	14.21	12.52	<u>54.37</u>	<u>54.07</u>
SimpleSTC	35.33	35.62	59.37	59.01	20.67	20.45	50.00	40.47
GHGA-Net	51.45	**51.91**	**68.75**	**68.03**	**47.17**	**47.13**	**58.74**	**57.63**
relative↑(%)	−2.13	1.47	4.77	3.93	4.22	5.46	8.04	6.58

data. Although the pre-training model has undergone a large amount of corpus training, there is still a gap in accuracy compared with our method. The results strongly prove the influential contribution of our global heterogeneous graph constructing based on the original documents information.

Table 3. Training cost compare with pre-training models. Evaluated in TNEWS dataset with 200 epochs.

Mode	Parameters	Hidden size	Layers	Times
BERT	102.28M	768	12	9 m 37 s
RoBERTa	102.28M	768	12	9 m 40 s
ERNIE	99.88M	768	12	5 m 02 s
GHGA-Net	0.605M	256	6	58.88 s

For training cost, we compare GHGA-Net with pre-training models. Table 3 shows the results. Our method has much fewer training parameters and less time consumption, but it achieves better performance. A lightweight structure makes GHGA-Net more efficient for real task and deployment.

4.4 Ablation Study

Recall that the proposed global heterogeneous graph and attention mechanism, we designed ablation experiments with different variants of GHGA-Net: (1)

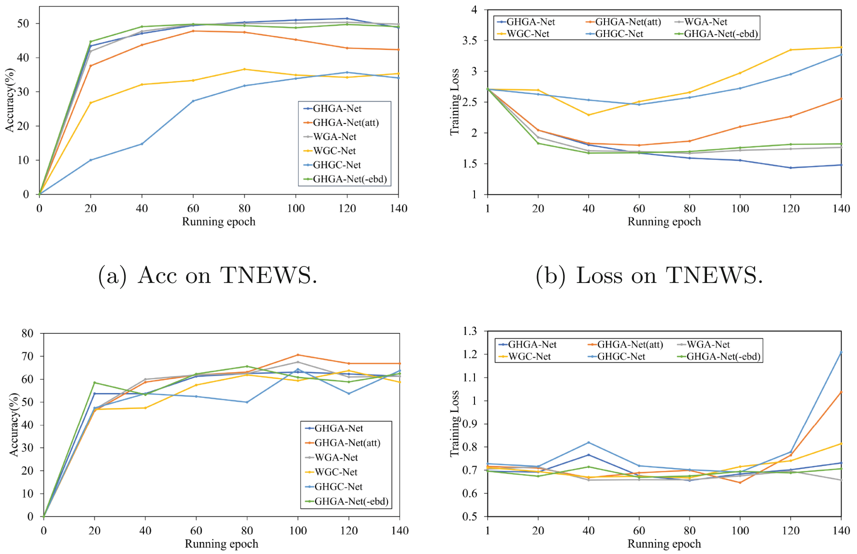
WGC-Net: without character features and attention layers, use GCN for training; (2) **HGC-Net:** without attention layers and use GCN for training; (3) **WGA-Net:** without character features; (4) **GHGA-Net(-ebd)**: using identity matrix instead of pre-trained vector in 2; (5) **GHGA-Net(fuse method)**: As mentioned in Chap. 3, besides concatenate operation in 9, we test the effect of linear interpolation and attention network for the fusion of features.

Table 4. Ablation Study (%) mesured on FewCLUE datasets. Trained under TrainAll set for 500 epoch. The - mark means unable to fit. The best results are marked in bold.

	TNEWS		EPRSTMT	
Method	ACC	F1	ACC	F1
WGC-Net	36.24	36.37	66.25	65.71
GHGC-Net	35.7	35.96	66.87	66.65
WGA-Net	50.36	51.08	67.5	66.61
GHGA-Net(-ebd)	49.82	49.90	68.12	67.66
GHGA-Net(linear)	–	–	46.25	31.62
GHGA-Net(att)	47.81	47.54	**70.62**	**70.28**
GHGA-Net(ours)	**51.45**	**51.91**	68.75	68.03

Table 4 lists the results, we can see the improvement of introducing pre-trained word vectors compared with initial encoding in both datasets. Figure 2(a) shows the significant effect of our proposed graph attention mechanism for graph representation learning. Compared with the graph convolution method, the accuracy rate is improved by more than 15%. Due to the fact that the simple convolution does not pay attention to all key category features. As can be seen from the loss curves in Fig. 2(b) and Fig. 2(d), with the increase of training rounds, the loss of ordinary convolution methods will rise, and the introduction of attention mechanism can effectively solve this problem. Among all the curves, our proposed GHGA-Net is the smoothest and also the most stable, which strongly proves that we have adopted the optimal method.

In terms of embedding fusion, the linear interpolation method has the worst performance, which indicates that the simple weighted average will lose the original information. As shown in Fig. 2(c), the attention-based fusion method achieves the best accuracy on the EPRSTMT dataset. Although the performance on the TNEWS dataset is slightly worse, it proves the feasibility of using neural network based methods to fuse text features. However, it cannot be ignored that with the increase of the number of training rounds, the accuracy rate of the att-fusion method has declined and the loss has increased, which may be caused by overfitting and needs further experiments in future research.



(a) Acc on TNEWS.

(b) Loss on TNEWS.

(c) Acc on EPRSTMT.

(d) Loss on EPRSTMT.

Fig. 2. Performance in the first 140 epoch training.

5 Conclusions

In this paper, we propose the GHGA-Net for Chinese STC without relying on pre-training. By constructing heterogeneous global graph, we can make full use of the unlabeled texts, and the finally feature fusion of character and word is more suitable for the classification task of Chinese text. Experiments results show that our method outperforms existed models on few-shot learning in Chinese STC scenario, especially in case of minimal training data. The additional ablation study strongly prove that our graph representation learning based on attention mechanism can effectively reduce the noise and highlight the key information. Despite those achievements, there are also some limitations to improve: (i) we have tested that remove some high frequency words in different domains may help reduce noise. (ii) we could create embedding by diagonal matrix for words out of vocabulary to capture rare semantics. (iii) radicals and some implied features of Chinese can be added to heterogeneous graph. (iv) we intend to adapt our hierarchical attention to transformer-like, which could further benefit the text feature learning. We will conduct in-depth research in future works.

Acknowledgement. This work was supported by the Research Funds for the Institute of Information Engineering, Chinese Academy of Sciences (No. BMKY2021B04, No. BMKY2023B04, No. E1R0141104).

References

1. Kim, Y.: Convolutional neural networks for sentence classification. Eprint Arxiv (2014)
2. Tao, H., Tong, S., Zhao, H., Xu, T., Jin, B., Liu, Q.: A radical-aware attention-based model for Chinese text classification. In: AAAI Conference on Artificial Intelligence (2019)
3. Wankhade, M., Rao, A.C.S., Kulkarni, C.: A survey on sentiment analysis methods, applications, and challenges. Artif. Intell. Rev. **55**, 5731–5780 (2022)
4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS (2016)
5. Hu, L., Yang, T., Shi, C., Ji, H., Li, X.: Heterogeneous graph attention networks for semi-supervised short text classification. ACM Trans. Inf. Syst. (TOIS) **39**, 1–29 (2019)
6. Ye, Z., Jiang, G., Liu, Y., Li, Z., Yuan, J.: Document and word representations generated by graph convolutional network and bert for short text classification. In: European Conference on Artificial Intelligence (2020)
7. Wang, Y., Wang, S., Yao, Q., Dou, D.: Hierarchical heterogeneous graph representation learning for short text classification. arXiv e-prints (2021)
8. Zheng, K., Wang, Y., Yao, Q., Dou, D.: Simplified graph learning for inductive short text classification. In: Conference on Empirical Methods in Natural Language Processing (2022)
9. Zhou, Y., Xu, B., Xu, J., Yang, L., Li, C., Xu, B.: Compositional recurrent neural networks for Chinese short text classification. 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 137–144 (2016)
10. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.H.: Hierarchical attention networks for document classification. In: North American Chapter of the Association for Computational Linguistics (2016)
11. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. ArXiv arXiv:1805.02023 (2018)
12. Xu, L., et al.: Fewclue: a Chinese few-shot learning evaluation benchmark. ArXiv arXiv:2107.07498 (2021)
13. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. ArXiv arXiv:1809.05679 (2018)
14. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., Bengio, Y.: Graph attention networks. ArXiv arXiv:1710.10903 (2017)
15. Ding, K., Wang, J., Li, J., Li, D., Liu, H.: Be more with less: hypergraph attention networks for inductive text classification. In: Conference on Empirical Methods in Natural Language Processing (2020)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. ArXiv arXiv:1810.04805 (2019)
17. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: Annual Meeting of the Association for Computational Linguistics (2019)
18. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach. ArXiv arXiv:1907.11692 (2019)

19. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Mining Text Data (2012)
20. Tamekuri, A., Nakamura, K., Takahashi, Y., Yamaguchi, S.: Providing interpretability of document classification by deep neural network with self-attention. J. Inf. Process. **30**, 397–410 (2022)
21. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)

KSRE-CNER: A Knowledge and Semantic Relation Enhancement Framework for Chinese NER

Jikun Dong[1], Kaifang Long[1], Jiran Zhu[1], Hui Yu[2](✉), Chen Lv[1],
Zengzhen Shao[3], and Weizhi Xu[1](✉)

[1] School of Information Science and Engineering, Shandong Normal University,
Jinan 250014, China
2021020969@stu.sdnu.edu.cn, xuweizhi@sdnu.edu.cn

[2] Business School, Shandong Normal University, Jinan 250014, China
huiyu0117@sdnu.edn.cn

[3] School of Data Science and Computing, Shandong Women's University,
Jinan 250002, China

Abstract. Chinese named entity recognition (CNER) constitutes a pivotal undertaking entailing the identification and classification of named entities present within Chinese text. Traditional approaches based on CNN and BiLSTM have been effective for sequence labeling tasks. Additionally, graph neural networks (GNNs) have shown promising results in improving Chinese NER performance by incorporating lexical knowledge. However, these methods may still face challenges in handling ambiguity and inaccurate boundary recognition in Chinese NER. To tackle these challenges, we propose a knowledge and semantic relation enhancement framework. This framework integrates N-gram information and lexical knowledge into a gated graph neural network (GGNN) to capture Chinese lexical information and reduce ambiguity. Moreover, we leverage the Transformer model to update the weight information of each node, aiming to eliminate the influence of incorrect matching lexicons and augment the model's capability to recognize entity boundaries. Comprehensive experiments conducted on diverse datasets, including Resume, CCKS2017, MSRA, and a self-constructed History dataset, substantiate that our proposed model attains comparable results.

Keywords: Natural language processing · Chinese named entity recognition · Gated graph neural network · Transformer · N-gram

1 Introduction

Named entity recognition (NER) [15] is a pivotal component within the domain of natural language processing (NLP). It has demonstrated extensive utility across a myriad of downstream applications, including relation extraction [13]

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 185–197, 2024.
https://doi.org/10.1007/978-981-99-7022-3_17

and information retrieval [2]. In contrast to English, Chinese text encounters greater challenges due to the lack of natural separators like spaces. This gives rise to issues such as ambiguous entity boundaries and intricate compositions.

To enhance the effectiveness of NER, Lample et al. [8] achieved great success in CNER by using a character-based approach. After that, Zhang and Yang [24] introduced the Lattice LSTM model, which can effectively integrate lexical information. Building upon this foundation, Liu et al. [11] introduced four different strategies to integrate lexical knowledge, and Gui et al. [5] and Ding et al. [3] used GNNs to fuse a large amount of lexical knowledge to assist in improving the performance in CNER task. However, these methods are not without limitations, including the challenge of precisely identifying entity boundaries when incorporating significant amounts of lexical knowledge, as well as the issue of ambiguity. For instance, as illustrated in Fig. 1, the character "长(Long)" is contained in the lexicons "市长(Mayor)" and "长江大桥(Yangtze River Bridge)". These ambiguous lexicons often share a common character, posing challenges for the model in accurately pinpointing entity boundaries through reliance on such lexical knowledge.

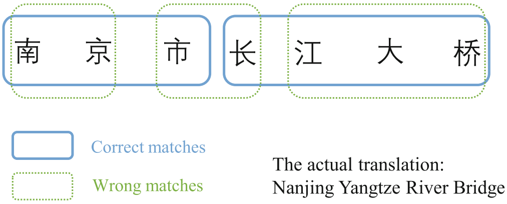


Fig. 1. Example of entity matching.

In response to the challenges posed by the introduction of incorrect word matching leading to ambiguity and the inaccuracies in entity boundary positioning observed in the aforementioned studies, we propose a Knowledge and Semantic Relation Enhancement framework for Chinese NER (KSRE-CNER) in this paper. The architecture utilizes GGNN to obtain character and lexical information in the sequence, utilizes Transformer to avoid the negative impact of incorrect lexicons, and utilizes BiLSTM to encapsulate contextual semantic understanding. The experimental outcomes underscore the remarkable performance achieved by the proposed model across three CNER datasets as well as a self-constructed dataset. Our contributions can be summarized as follows:

1. We propose the KSRE-CNER model, which can effectively capture contextual semantic information from sequences and mitigate the adverse effects of introducing erroneous lexicons, thus eliminating ambiguity in CNER.
2. We use Transformer to focus on important features and augment the model's capacity to accurately locate boundaries.

3. The conducted experiments showcase the proposed method's achievement of exceptionally satisfactory performance. Furthermore, ablation studies indicate that the proposed model effectively integrates lexical knowledge and contextual semantic information.

2 Related Work

With the progression of NLP, NER methodologies have evolved through three distinct developmental stages: (1) rule-based and dictionary-based approaches; (2) methodologies founded on statistical machine learning; (3) approaches rooted in deep learning.

Early NER methodologies predominantly revolved around rules and dictionaries. Rule-based techniques predominantly hinged on language experts to manually formulate rules, such as selecting punctuation marks, keywords and central words as feature construction rule templates, and using patterns and string matching as the primary means. This method is relatively simple and applicable, and many vocabularies can be identified based on existing dictionaries and rules [14,20]. Nonetheless, the limitations inherent in this rule-based approach are evident. It not only requires a huge amount of human labor, but also cannot be easily extended to other entity types or datasets.

Statistical machine learning-based approaches encompass a range of methodologies, notably the hidden markov model (HMM) [25], conditional random field (CRF) [18], and support vector machine (SVM) [7]. These models rely more on the feature selection of text. Selecting impactful features from the text to construct feature vectors, computing label scores using these feature vectors, and ultimately determining the optimal label sequence for the sentence are all imperative tasks. The introduction of statistical methods into NER also has some shortcomings. Machine learning models demand rigorous feature engineering, a process with stringent prerequisites. The quality of feature engineering significantly impacts the model's efficacy.

The development of neural networks has enabled huge performance improvement in NER. Deep learning approaches have demonstrated superior performance in comparison to traditional machine learning methods. The end-to-end BiLSTM-CRF model [6] is a representative and commonly used structure in NER. The Lattice LSTM [24] efficiently encodes both the character sequence and latent lexicons aligned with the dictionary, thereby effectively leveraging Chinese lexical insights. Building upon the foundation of the Lattice LSTM, the Flat [10] and NFLAT [22] utilize positional encoding lexical information and make good use of the masking mechanism of the Transformer [21]. In addition, methods [4,26] based on convolutional neural network (CNN) of Chinese NER can use a rethinking mechanism to integrate lexical information. [12] incorporate lexical dictionaries into character representations to improve model performance. Utilizing a graph structure [3] to capture lexicon information can fully use lexicon information while disambiguating.

3 Methodology

The objective of the CNER task is to discern and categorize entities alluded to in a provided sentence $S = (s_1, s_2, \ldots, s_n)$ into predefined label categories $Y = (y_1, y_2, \ldots, y_n)$. Figure 2 illustrates the comprehensive architecture of the proposed KSRE-CNER model, encompassing four principal modules: a) feature encoding module; b) graph module; c) Transformer module; d) information fusion and decoding module.

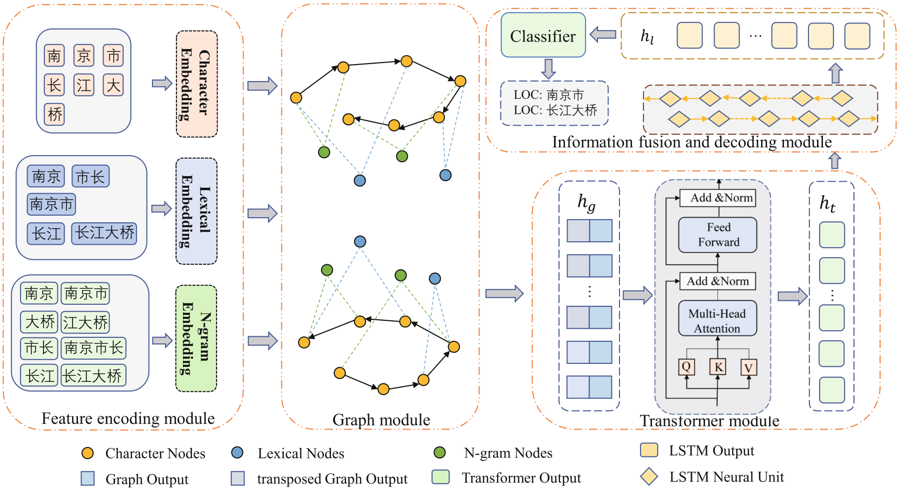


Fig. 2. The overall architecture of the KSRE-CNER model.

3.1 Feature Encoding Module

Character Embedding: Word2Vec is a widely used tool for word embedding representation in various NLP tasks. Here, we associate each character in the sentence S with a corresponding word vector from Word2Vec, thereby establishing the initial vector representation for the character node.:

$$h_g^0 = \left[(T^c)^T, \left(T^{bi}\right)^T \right]^T \tag{1}$$

where T^c and T^{bi} represent the lookup table of characters and bigram embedding table [1], respectively.

Lexical Embedding: The previous method [3,11,24] proved that using Chinese lexical knowledge can improve the performance. To fully utilize lexical knowledge, we employ both forward maximum matching (FMM) algorithm and backward maximum matching (BMM) algorithm to ascertain the lexical knowledge associated with each character. For instance, for a sentence "南京市长江大桥(Nanjing Yangtze River Bridge)", the character "长(Long)" finds the lexical knowledge "长江(Yangtse River)", "长江大桥(Yangtze River Bridge)" through FMM, and finds the lexical knowledge "市长(Mayor)" through BMM. Therefore, the lexical knowledge matched by the character "长(Long)" includes "长江(Yangtse River)", "长江大桥(Yangtze River Bridge)", and "市长(Mayor)". After that, we obtain the embedding of the lexicons to represent the initial state of the lexical node.

$$h_g^0 = T^v \tag{2}$$

where T^v is lookup table for the lexical node represents.

N-gram Embedding: Zhang and Yang [24] have demonstrated that N-gram knowledge is very effective in extracting boundary information of entities in sentences. Therefore, we enhance the sensitivity of the model to entity boundary information in the sentence by introducing the concept of frequency-based filtering in N-gram knowledge. First, we obtain 2-gram, 3-gram, and 4-gram information for all sentences in the dataset. Subsequently, we tally the frequency of each N-gram to compile the N-gram lexicons. For each sentence, we extract N-gram information by setting different thresholds. For example, for 4-gram, if the number of times that a certain character's 4-gram information appears in the N-gram lexicons is greater than 3, we keep it, otherwise we discard it. Similarly, we obtain the embedding of the N-gram lexicons to represent the initial state of the N-gram nodes.

$$h_g^0 = T^u \tag{3}$$

where T^u is lookup table for the N-gram node represents.

3.2 Graph Module

In this subsection, we first describe how to construct the nodes and edges of the graph, and then describe our approach in detail.

Nodes: As depicted in Fig. 2, there exist three different node types, namely character nodes, lexical nodes, and N-gram nodes. The yellow solid circles represent character nodes, which are intended to represent the character features of the sentence. The blue solid circles signify lexical nodes, while the green solid circles represent N-gram nodes.

Edges: Similarly, the graph encompasses three distinct types of edges: a black edge connecting character nodes, a blue edge connecting character nodes to word nodes, and a green edge connecting character nodes to N-gram nodes

Construct Graph: Based on our observation, the lexical knowledge and N-gram knowledge matched to the sentences may contain duplicates, which can augment the model's capability to learn the boundary information of entities in the sentences. In addition, the non-duplicated lexical information can also improve the model's generalization capacity.

Specifically, we construct a directed graph $G := (V, E, L)$. $V = \{V_c, V_v, V_u\}$ represents the collection of nodes, where V_c, V_v, and V_u denote character nodes, lexicon nodes, and N-gram nodes, respectively. E and L represent the collection of edges and their labels. Each edge in E carries a label signifying the connection between different nodes. For label set $L = \{L_c, L_v, L_u\}$, the label L_c is allocated to the edges connecting the characters. L_v is allocated to edges connecting characters and lexicon. And L_u is allocated to edges connecting characters and N-gram dictionary. In the given graph structure, we use adapted GGNN [3] to learn the weighted combination of the lexical dictionary and N-gram dictionary, and update the node information. We assign a trainable contribution coefficient $\beta_c, \beta_1, \beta_2, \ldots, \beta_{\mathrm{k}}$ (where k represents the number of all lexicons and N-gram entities matched by characters) to each edge. Subsequently, we broaden the scope of the adjacency matrix A to encompass edges characterized by diverse labels. This extended adjacency matrix A is utilized to retrieve neighboring node states at each stage. The contribution coefficients are further translated into edge weights within A via the application of a sigmoid activation function.

$$\alpha_c, \alpha_1, \alpha_2, \ldots, \alpha_k = \sigma\left(\beta_c, \beta_1, \beta_2, \ldots, \beta_k\right) \tag{4}$$

The GGNN employs GRU to transmit and update the hidden information of nodes within the graph. The node formula is then updated in the following manner:

$$H = \left[h_1^{t-1}, h_2^{t-1}, \ldots, h_{|V|}^{t-1}\right]^\top \tag{5}$$

$$a_g^t = \left[\left(HW_1\right)^T, \left(HW_2\right)^T, \ldots, \left(HW_{|L|}\right)^T\right]^T A_g^T + b \tag{6}$$

$$z_g^t = \sigma\left(W^z a_g^t + U^z h_g^{t-1}\right), r_g^t = \sigma\left(W^r a_g^t + U^r h_g^{t-1}\right) \tag{7}$$

$$\check{h}_g^t = \tanh\left(W a_g^t + U\left(r_g^t \odot h_g^{t-1}\right)\right), \tag{8}$$

$$h_g^t = \left(1 - z_g^t\right) \odot h_g^{t-1} + z_g^t \odot \check{h}_g^t \tag{9}$$

At each time step t, the vector representation H is composed of the concatenated vector representations of all nodes at time step $t-1$. The interaction of the g-th node with its adjacent nodes is symbolized as a_g^t, where A_g denotes the row vector associated with the g-th node in the adjacency matrix A of the graph. The trainable parameters W and U are used to compute the interaction between the nodes. The output z_g^t controls the forgotten information, r_g^t controls the freshly incorporated information, and h_g^t embodies the ultimate updated node state at time step t.

Furthermore, inspired by Gui [5], we construct a transposed graph. Similarly, we obtain the hidden information of the all nodes. Ultimately, the ultimate vector representation of the node is determined in the subsequent manner:

$$h_g = (\overrightarrow{h_g}; \overleftarrow{h_g}) \tag{10}$$

3.3 Transformer Module

To capture global sequence information and extract essential features from the text sequence for the purpose of bolstering entity boundary information, we utilize the encoder module of Transformer [21]. As shown in Fig. 2, within the architecture of the Transformer, two sub-layers are integral components: the multi-head Attention mechanism and the feedforward neural network (FFNN). The specific calculation formulas are outlined below:

$$h_a = \text{LayerNorm}(h_g + \text{MHAttention}(Q, K, V)) \tag{11}$$

$$h_t = \text{ LayerNorm } \left(h_a + \text{FFNN}\left(h_a\right)\right) \tag{12}$$

The *MHAttention* corresponds to the multi-head Attention mechanism integrated within the Transformer encoder module. This extraction aims to amplify the precision of entity boundary information. Here, Q, K, and V denote the query, key, and value vectors, respectively. These vectors are derived through linear transformations of the vector h_g. The output of the first layer of *MHAttention* is labeled as h_a, while the ultimate output of the Transformer is designated as h_t.

3.4 Fusion and Decoding Module

To better integrate context information and obtain the final labeled result for the sequence, we input h_t into the BiLSTM-CRF model [6] for fusion and decoding.

$$h_{l(i)} = \left(\overrightarrow{\text{LSTM}}\left(h_{t(i)}, \overrightarrow{h_{l(i-1)}}\right); \overleftarrow{\text{LSTM}}\left(h_{t(i)}, \overleftarrow{h_{l(i-1)}}\right)\right) \tag{13}$$

where ";" signifies the concatenation operation. Ultimately, the representation of the character sequence can be denoted as $h_l = \left\{h_{l(1)}, \ldots, h_{l(i)}, \ldots, h_{l(n)}\right\}$. After that, h_l is fed into the CRF layer, which is responsible for assigning labels to each word in order to generate the label sequence $Y = (y_1, \ldots, y_i, \ldots, y_n)$.

$$y_i = \text{argmax}_{y'} P\left(y' \mid h_{l(i)}\right), L_{\text{loss}} = -\sum_{i=1}^{n} \log P\left(y_i \mid h_{l(i)}\right) \tag{14}$$

where y' denotes all possible label sequences, $P\left(y_i \mid h_{l(i)}\right)$ is the probability of label sequence y' given $h_{l(i)}$. L_{loss} denotes the loss function.

4 Experiments

4.1 Datasets

To assess the impact of the proposed KSRE-CNER model, we perform experiments on three publicly available datasets as well as our self-constructed datase, including Resume [24], MSRA [9] , CCKS2017[1], and History[2]. The Resume, MSRA, CCKS2017 and History datasets are composed of Chinese resumes, news, biomedical and historical data, respectively. Table 1 presents the statistics regarding the number of sentences and characters within each dataset. Furthermore, our dataset includes 9 distinct label categories, including organization, location, date, person, salutation, appellation, event, army, and place of affiliation. To facilitate specific entity identification, we partition the History dataset into two classifications. The initial category comprises nine tag types, denoted as History-9types. Meanwhile, the second category, History-3types, encompasses entities related to location, appellation, and event.

Table 1. Statistics of datasets.

Datasets	Type	Train	Test	Dev
Resume	Character	124.1K	15.1K	13.9K
	Sentence	3.8K	0.48K	0.46K
MSRA	Character	2169.9K	172.6K	–
	Sentence	46.4K	4.4K	–
CCKS2017	Character	200.0K	33.6K	31.8K
	Sentence	5.9K	1.09K	0.82K
History	Character	289.1K	29.9K	30.9K
	Sentence	8.9K	0.81K	0.97K

4.2 Implementation Details and Evaluation Metrics

Throughout the training process, we conduct 100 epochs, with each epoch involving a batch size of 10. The learning rate is configured at 0.001, while the word embedding dimension is maintained at 50. Moreover, we use the SGD optimizer. Additionally, we utilize precision (P), recall (R), and F1 score (F1) as evaluation metrics to assess the performance of our model on these datasets.

[1] https://www.biendata.xyz/competition/CCKS2017_2/.
[2] https://github.com/BIG-SMILE/history_dataset_ner.

4.3 Main Results

To assess our model's effectiveness, we benchmark it against a baseline, BiLSTM-CRF [6]. Additionally, we contrast our model with several recently proposed models, outlined as follows. (1) Lattice [24] significantly enhances the performance of the CNER task by integrating word knowledge into characters through a lattice structure for the first time. (2) WC-LSTM [11] provides four strategies to fuse word knowledge. (3) Multi-digraph [3] is a model to fuse lexical knowledge by adapting a GGNN. (4) SoftLexicon [12], as introduced by Ma et al., presents a unique strategy for fusing lexical knowledge.(5) LR-CNN [4] enables the model to have the ability to re-select words through a rethinking mechanism. (6) CAN-NER [26] uses a convolutional attention to enhance the performance of CNER. (7) TENER [23] adapts the Transformer encoder to model both character-level and word-level attributes, thus enhancing NER performance. (8) FLAT [10] enhances NER with flat lattices, while NFLAT [22] further reduces memory usage and improves efficiency by decoupling lexicon fusion and context encoding. (9) The Locate and Label model [16] effectively recognizes entities by leveraging boundary information and partially matched spans, surpassing previous methods. (10) The Sequence-to-Set model [19] captures dependencies between entities and achieves good performance on NER. (11) PIQN [17] extracts entities in parallel using learnable instance queries, outperforming previous NER models.

As evident from Tables 2, 3, and 4, our model shows excellent results in contrast to the above models. One potential explanation is that our model proficiently identifies entity boundaries within the sentences. Simultaneously, our model can sufficiently capture the semantic information and contextual knowledge in the sentences.

Table 2. Performance on Resume.

Models	Resume		
	P	R	$F1$
Baseline [6]	93.73	93.44	93.58
Lattice [24]	94.81	94.11	94.46
CAN-NER [26]	95.05	94.82	94.94
WC-LSTM [11]	95.14	94.79	94.96
LR-CNN [4]	95.37	94.84	95.11
TENER [23]	–	–	95.00
FLAT [10]	–	–	95.45
NFLAT [22]	95.63	95.52	95.58
KSRE(ours)	**95.66**	**95.59**	**95.62**

Table 3. Performance on MSRA.

Models	MSRA		
	P	R	$F1$
Baseline [6]	91.28	90.62	90.95
Locate and Label [16]	92.20	90.72	91.46
Sequence-to-Set [19]	93.21	91.97	92.58
TENER [23]	–	–	92.74
CAN-NER [26]	93.53	92.42	92.94
Lattice [24]	93.57	92.79	93.18
PIQN [17]	**93.61**	93.35	93.48
KSRE(ours)	93.17	**93.92**	**93.55**

Table 4. Performance on History and CCKS2017 datasets.

Models	History-9types			History-3types			CCKS2017		
	P	R	$F1$	P	R	$F1$	P	R	$F1$
Baseline [6]	76.01	60.68	67.48	76.15	50.45	60.69	88.45	87.35	87.90
WC-LSTM [11]	82.43	68.48	74.81	**83.19**	61.15	70.48	88.96	87.33	88.14
Multi-digraph [3]	75.31	71.70	73.46	76.14	59.36	66.71	89.50	**88.40**	88.94
SoftLexicon [12]	**82.65**	70.11	75.86	82.47	67.13	74.02	89.67	87.23	88.43
KSRE(ours)	80.57	**73.09**	**76.65**	76.57	**73.87**	**75.20**	**89.92**	88.37	**89.14**

4.4 Ablation Study

To scrutinize the efficacy of lexical knowledge and various modules within our framework, we undertake a comparison between the complete model and its ablation variants.

As illustrated in Table 5, the utilization of external knowledge contributes to the enhanced performance of our model. Specifically, on the History-9types dataset, when the lexical knowledge is not used, **w/o Lexicon** drops 2.30% F1; when the N-gram knowledge is not used, **w/o N-gram knowledge** drops 1.56% F1; when both lexical knowledge and N-gram knowledge are not used, **w/o Lexicon+N-gram** drops 4.02% F1. On the History-3types dataset, when the lexical knowledge is not used, **w/o Lexical** drops 1.49% F1; when the N-gram knowledge is not used, **w/o N-gram** drops 0.24% F1; When both lexical knowledge and N-gram knowledge are not used, **w/o Lexicon+N-gram** drops 1.90% F1. These results suggest that integrating lexical knowledge and N-gram knowledge has a positive impact on our model's performance.

Table 5. Ablation study of the influence of lexical knowledge on model performance. "w/o" means to remove a component.

Models	History-9types			History-3types		
	P	R	$F1$	P	R	$F1$
KSRE(ours)	**80.57**	73.09	**76.65**	76.57	73.87	**75.20**
w/o Lexicon	75.32	**73.41**	74.35	74.57	72.88	73.71
w/o N-gram	77.13	73.16	75.09	73.39	73.59	74.96
w/o Lexicon+N-gram	75.53	69.95	72.63	71.30	**75.42**	73.30

Table 6. Ablation study of the influence of different components on model performance. "w/o" means to remove a component.

Models	CCKS2017			Resume		
	P	R	$F1$	P	R	$F1$
KSRE(ours)	**89.92**	88.37	**89.14**	**95.66**	**95.59**	**95.62**
w/o Transformer	89.17	87.61	88.38	94.15	93.87	94.01
w/o LSTM	87.61	**89.36**	88.48	93.91	94.60	94.25
w/o Transformer+ LSTM	87.41	87.76	87.59	93.45	93.62	93.53

As depicted in Table 6, On the CCKS2017 dataset, the absence of the Transformer module (**w/o Transformer**) results in a decline of 0.76% in F1. Similarly, the absence of the LSTM module (**w/o LSTM**) leads to a decrease of 0.66% in F1. When both the Transformer module and LSTM module are excluded (**w/o Transformer + LSTM**), a more substantial drop of 1.55% in F1 is observed. On the Resume dataset, the exclusion of the Transformer (**w/o Transformer**) leads to a decrease of 1.61% in F1 score, while the omission of the LSTM module (**w/o LSTM**) results in a reduction of 1.37%. The exclusion of both the Transformer module and LSTM module (**w/o Transformer + LSTM**) leads to a 2.09% decrease in F1. These findings affirm the crucial roles played by both the Transformer module and the LSTM module within our framework.

5 Conclusion

In this study, we propose a knowledge and semantic relation enhancement framework for Chinese NER. Lexical information and N-gram information are introduced into the GGNN to enhance the model's capability to eliminate ambiguity. In addition, we use Transformer and BiLSTM to enhance boundary and context representation. The results obtained from experiments on the Resume, MSRA, CCKS2017, and self-constructed History dataset affirm the efficacy of our proposed approach. Ablation studies show that the lexical information, N-gram information, Transformer and BiLSTM are beneficial for CNER.

Acknowledgements. This work was supported in part by Natural Science Foundation of Shandong Province (No. ZR2022MF328, No. ZR2019LZH014 and ZR2021MF059), and in part by National Natural Science Foundation of China (No. 61602284 and No. 61602285).

References

1. Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.J.: Long short-term memory neural networks for Chinese word segmentation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1197–1206 (2015)

2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1: Long Papers, pp. 167–176 (2015)
3. Ding, R., Xie, P., Zhang, X., Lu, W., Li, L., Si, L.: A neural multi-digraph model for Chinese ner with gazetteers. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1462–1467 (2019)
4. Gui, T., Ma, R., Zhang, Q., Zhao, L., Jiang, Y.G., Huang, X.: CNN-based Chinese ner with lexicon rethinking. In: IJCAI, pp. 4982–4988 (2019)
5. Gui, T., et al.: A lexicon-based graph neural network for Chinese NER. 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. 1040–1050 (2019)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. Ju, Z., Wang, J., Zhu, F.: Named entity recognition from biomedical text using SVM. In: 2011 5th International Conference on Bioinformatics and Biomedical Engineering, pp. 1–4. IEEE (2011)
8. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
9. 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 (2006)
10. Li, X., Yan, H., Qiu, X., Huang, X.J.: Flat: Chinese NER using flat-lattice transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6836–6842 (2020)
11. Liu, W., Xu, T., Xu, Q., Song, J., Zu, Y.: An encoding strategy based word-character LSTM for Chinese NER. 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. 2379–2389 (2019)
12. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.J.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5951–5960 (2020)
13. Mooney, R., Brew, C., Chien, L.F., Kirchhoff, K.: Proceedings of human language technology conference and conference on empirical methods in natural language processing. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005)
14. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
15. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
16. 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 the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, vol. 1: Long Papers), pp. 2782–2794 (2021)
17. Shen, Y., et al.: Parallel instance query network for named entity recognition. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 947–961 (2022)

18. Skeppstedt, M., Kvist, M., Nilsson, G.H., Dalianis, H.: Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. J. Biomed. Inform. **49**, 148–158 (2014)
19. Tan, Z., Shen, Y., Zhang, S., Lu, W., Zhuang, Y.: A sequence-to-set network for nested named entity recognition. arXiv preprint arXiv:2105.08901 (2021)
20. Tsuruoka, Y., Tsujii, J.: Boosting precision and recall of dictionary-based protein name recognition. In: Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, vol. 13. pp. 41–48. Citeseer (2003)
21. Vaswani, A., et al.: Attention is all you need. Adv. Neural. Inf. Process. Syst. **30**, 1–11 (2017)
22. Wu, S., Song, X., Feng, Z., Wu, X.: Nflat: non-flat-lattice transformer for Chinese named entity recognition. arXiv preprint arXiv:2205.05832 (2022)
23. Yan, H., Deng, B., Li, X., Qiu, X.: Tener: adapting transformer encoder for named entity recognition. arXiv preprint arXiv:1911.04474 (2019)
24. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 1554–1564 (2018)
25. Zhou, G., Su, J.: Named entity recognition using an hmm-based chunk tagger. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 473–480 (2002)
26. Zhu, Y., Wang, G.: Can-ner: convolutional attention network for Chinese named entity recognition. 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. 3384–3393 (2019)

Low-Frequency Aware Unsupervised Detection of Dark Jargon Phrases on Social Platforms

Limei Huang, Shanshan Wang, Changlin Liu, Xueyang Cao, Yadi Han, Shaolei Liu, and Zhenxiang Chen(✉)

School of Information Science and Engineering, University of Jinan, Jinan, China
{lmhuang,xueyangcao}@stu.ujn.edu.cn, {ise_wangss,zxchen}@ujn.edu.cn,
15253464198@163.com, yadi@163.com, lcl_go@163.com

Abstract. With the development of the Internet, the number of people communicating on social platforms has soared, which means that it is crucial for platform moderators to review and remove illegal content to create a clean network environment for users. However, identifying such content becomes complex due to the use of dark jargons. These jargons are seemingly innocent or newly coined words and phrases, such as "coke" for cocaine or "vanilla sky" for synthetic cathinone, to convey illegal meanings, aiming to evade detection by moderators. Existing methods primarily focus on detecting dark jargons at the word level, yielding commendable results. However, given the prevalence of phrase-level dark jargons in the context, relying solely on word-level detection can introduce ambiguity. For example, "black" is not a dark jargon, but "black bart" is a dark jargon. As a result, there is a growing interest in developing techniques specifically targeting phrase-level dark jargon detection. Unfortunately, such efforts are relatively limited, potentially resulting in the oversight of numerous low-frequency dark jargon phrases. To tackle this challenge, we propose the Low-Frequency Aware Dark Jargon Phrases Detection (DJPD) model. Our approach centers around finding a noun phrasal attention map pattern based on Transformer that enhances the perception of low-frequency phrases, enabling the selection of candidate dark jargon phrases. Subsequently, the candidate dark jargon phrases' sentence-level context is analyzed to detect dark jargon phrases. Remarkably, our model achieves a significant 84.66% improvement in F1-score compared to the current state-of-the-art method for dark jargon phrase detection in the corpus.

Keywords: Dark jargon phrases · Low-frequency awareness · Unsupervised learning · Attention maps · Context representation

1 Introduction

With the development of the Internet, large social platforms begin to employ a large number of moderators to review and delete content related to cyber

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 198–209, 2024.
https://doi.org/10.1007/978-981-99-7022-3_18

crimes, so as to ensure that the content on the platform conforms to relevant laws and regulations. However, according to Forbes [8], Facebook makes 300,000 content moderation mistakes every day for some reasons. First and foremost, cybercrime-related underground industries, such as drugs, pornography, weapons, etc., engage in online transactions while evading moderation filters. As a result, they constantly evolve their use of dark jargon, such as referring to cocaine as "coke" to elude detection [5]. Dark jargon refers to the use of seemingly innocuous words, such as "coke" for cocaine, or newly coined expressions, such as "vanilla sky" for synthetic meth, which carry illicit connotations and is deliberately employed to evade the scrutiny of moderators [19]. This hidden and constantly changing dark jargon has brought great challenges to content moderation on social platforms.

The current detection methods of dark jargon are divided into word level(such as "weed" for marihuana) and phrase level(such as "black bart" for marijuana):

(1) Most of these detection methods focus on the automatic detection of word level [6,10,16–20,22], and have achieved relatively good research results. However, the minimal expression unit of a large number of dark jargon in the context is always the phrase, and the automatic detection of unigram words will lead to semantic ambiguity, which will cause the problem of missed detection or false detection of dark jargon. For example, "oil" is not a dark jargon, but "cbd oil" is one. Here, a dark jargon phrase is a sequence of words of arbitrary length that appear continuously in a sentence and contain illegal content, forming a complete semantic unit in a specific context [15]. Only detecting dark jargon at the phrase level can capture the contextual semantic information of minimal expression units, effectively detect dark jargon, and help moderators improve the efficiency of purifying the content of social platforms.

(2) To the best of our knowledge, there is a relative lack of research on automated phrase-level dark jargon detection. Moreover, previous work relies on general domain methods for selecting high-quality phrases [21]: Frequent n-grams are usually used to find candidate phrases based on the corpus. However, this scheme is not suitable for detecting dark jargon phrases for the following reasons: (1)Due to the moderator detecting and filtering the dark jargon, criminals must constantly evolve the dark jargon to conduct illegal transactions online, resulting in most of the dark jargon being low-frequency phrases. (2)And the frequency threshold of candidate phrases obtained by using the high-frequency n-gram method based on a corpus, many dark jargons at the low-frequency phrase level have been missed. Therefore, how to design a model of low-frequency dark jargon phrase perception to improve the accuracy of dark jargon phrase detection has become an urgent problem to be solved.

In order to solve the above problems, this paper designs a low-frequency aware Dark Jargon Phrases Detection (DJPD) model. The model consists of three modules: candidate dark jargon phrase selection module, phrase-level context representation module, and dark jargon phrase detection module. Firstly, we transform phrase selection into image classification to select candidate dark jargon phrases. Secondly, the context representation module uses the Black-BERT model to generate the high-quality contextual representation of each complete semantic unit. Thirdly, the dark jargon phrase detection module based on cross-representation comparison detects the dark jargon phrases. Finally, our model improves F1-score by 84.66% compared with the state-of-the-art model in dark jargon phrase detection.

The main contributions of this study are as follows:

- We propose a selection module of candidate dark jargon phrases to fill the gap of missing low-frequency dark jargon phrases by current detection methods.
- We innovatively fine-tune a phrase-based pre-trained BERT model called the Black-BERT model to generate a high-quality contextual representation.
- We carry out extensive experiments on our dataset and show our model has superior performance in detecting dark jargon phrases on social platforms.

2 Related Work

Dark jargon detection is a relatively new research field that is still in its infancy. Due to the continuous evolution of dark jargon, existing researches on dark jargon detection are divided into word level and phrase level.

Word-level dark jargon detection has achieved good results, including supervised, semi-supervised, and unsupervised learning.

For example, Wang et al. [18] and Li et al. [10] proposed a supervised method to detect dark jargon. Nevertheless, Wang et al. 's and Li et al.' s methods require a large number of data annotations, and their detection results are extremely dependent on the quality of data annotations.

Another group of related studies [17,19] proposed a semi-supervised method. [17] extracts seven new features of Chinese jargon, used transfer learning to improve the quality of word vectors, and finally used statistical outlier detection to determine whether a word was Chinese dark jargon. However, the results depend on the limited labeled samples for model training, which leads to poor generalization ability of the model. In addition, Yang et al. [19] adopted a different idea to capture the Chinese dark jargon by searching for seed keywords and crawling the pages of search engine alerts. However, the dark jargon it captures depends on the alarm of search engines.

Next, we introduce four unsupervised approaches [6,16,20,22]. Takuro et al. [16] and Ke et al. [6] use an unsupervised cross-corpus comparison method to detect Chinese dark jargon. SCM improved word2vec so that it can compare the difference of word vectors of the same words in two corpora (i.e., legal corpus and illegal corpus), to detect dark jargon used for illegal purposes. In addition, Zhu et al. [22] proposed a different idea, a self-supervised way, using the BERT masked language model. Specifically, by analyzing words in their sentence-level context detect dark jargon. However, as shown in the discussion of SCM [20] and Euphemism Detection [22], these two methods only perform word-level dark jargon detection and do not support phrase-level one.

Phrase-level dark jargon detection is scarce. EPD [21] uses the statistical feature-based method Autophrase [15] to detect phrases and then uses the word vector generated by Word2vec to filter out the candidate dark jargon phrases related to the underground industry. Finally, the masked language model of SpanBERT is used to rank each candidate's dark jargon phrase by predicting the weighted sum of its mask.

In conclusion, the existing methods for detecting jargon have many common limitations: most of them rely on high-frequency n-gram statistical features to detect dark jargon phrases. However, for the sake of regulators, most of the jargon phrases are low-frequency, which have been missed by previous methods.

3 Methodology

In order to detect dark jargon phrases on social platforms more efficiently, we propose the DJPD model, which has three modules, as shown in Fig. 1: (1) selecting candidate dark jargon phrases, (2) generating the contextual representation of phrases, (3) detecting dark jargon phrases.

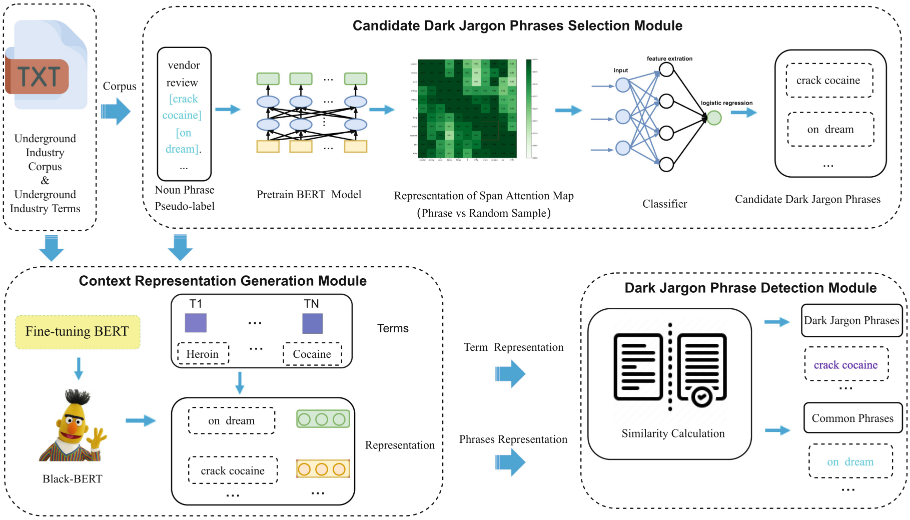


Fig. 1. Low-frequency aware unsupervised dark jargon phrases detection model.

3.1 Candidate Dark Jargon Phrases Selection Module

3.1.1 Candidate Dark Jargon Phrases Pseudo-label Generation

In this step, We collect high-quality candidate dark jargon phrase pseudo-labels from underground industry corpus[1] in an unsupervised manner, which will

[1] https://www.reddit.com/r/bigquery/comments/3cej2b/17_billion_reddit_comments_loaded_on_bigquery/.

be training used for candidate dark jargon phrase classifier f_θ, parameterized by θ. We treat the whole corpus as D, which consists of individual sentence *[$token_1$,...,$token_N$]*. Here, N is the length of the sentence. Firstly, we regard the largest continuous span that appears more than once in the corpus as candidate dark jargon phrase pseudo-labels L_{i_j}. Here L_{i_j} is the j-th phrase pseudo-labels in the i-th sentence. In other words, we select the largest continuous the j-th span *[$token_{l_j}$,...,$token_{r_j}$]* from the i-th sentence *[$token_{1_i}$,...,$token_{N_i}$]*, where $token_l$ is the leftmost token of the phrases, $token_r$ is the rightmost token of the phrases, i is the i-th sentence, and j is the j-th phrase. The "largest" here refers to the phrase represented as a complete semantic unit in a sentence [15]. Secondly, to ensure the completeness of dark jargon phrases, we only keep span *[$token_l$,...,$token_r$]* that is not a sub span of another span. Thirdly, to further improve the informativeness of candidate dark jargon phrase pseudo-labels L_{i_j}, that is, high-quality candidate dark jargon phrases should be words specific to the field of underground industry. We use the stop words list [15] to filter out the span *[$token_l$,...,$token_r$]* containing stop words. Fourthly, most of the candidate's dark jargon phrases are noun phrases. Here, noun phrases refer to pronouns or nouns plus simple modifiers, like adjectives and demonstratives [4]. we apply Natural Language Toolkit (NLTK) [11] to perform part-of-speech tagging, and then use part-of-speech rules of noun phrases in the form of regular expressions, as shown in Eq. 1, to select candidate dark jargon phrase pseudo-labels L_{i_j}.

$$R = \text{``}NP: <JJ> * <NN.*> +\text{''} \tag{1}$$

where R defines the part-of-speech rules of noun phrases in the form of regular expressions, NP represents the name of these rules, $<JJ>*$ refers to zero or more adjectives, and $<NN.*>+$ indicates at least one noun.

Finally, to ensure the same number of positive and negative samples, which is used for training candidate dark jargon phrase classifier f_θ, parameterized by θ, we add the above-obtained effective span *[$token_l$,...,$token_r$]* to the positive sample library S^p_{max}, while randomly draw the same number spans from the remaining spans to the negative sample library S^n_{max}. Here, p is the positive sample, max refers to "the largest", and n refers to the negative sample.

3.1.2 Span Attention Distribution Representation Extraction

In order to increase the generalization ability of the Candidate Dark Jargon Phrases Selection Module, the testing phase is not assisted by word frequency, and the perception ability of low-frequency dark jargon phrases is enhanced to select high-quality candidate dark jargon phrases. We are inspired by [7] showing that patterns often appear in the self-attention heatmaps of pre-trained BERT models, which indicate that the distribution of attention across words in the same phrase is relatively similar. Therefore, we propose to leverage pre-trained BERT models to obtain the attention distribution representation of the span labeled by Sect. 3.1.1 in the positive sample library S^p_{max} and the negative sample library S^n_{max}.

Firstly, the sentence *[token*$_1$*,...,token*$_N$*]* is input into the pre-trained BERT model as units. Then, We map the sentence *[token*$_1$*,...,token*$_N$*]* to their corresponding integer encoding *ids=[id*$_1$*,...,id*$_N$*]*. And, we format *ids* by adding bos_token_id before the integer encoding and pad_token_id at the end and making *ids* have the same length in each batch(using the longest sentence as the criterion in each batch) using pad_token_id padding *ids*, as shown in Eq. 2. Here, bos_token_id is a special identifier used to indicate the beginning of a sequence. And, pad_token_id is a special identifier used to indicate a padding tag. And, the pre-trained BERT model$g$ represents the "allenai/cs_roberta_base" model without fine-tuning, as shown in Eq. 2.

$$B = g(ids, atten\_mask, out\_atten, ret\_dict) \tag{2}$$

where *atten_mask* is the binary mask used to control the attention mechanism, in which "1" is converted by a real *ids* and "0" is converted by a pad_token_id; *out_atten* indicates whether to output the attention distribution representation. Whether the BERT model g will return the results in dictionary form is indicated by the variable *ret_dict*. And, B is the output results of the BERT model.

Accessing values through dictionary retrieval (i.e. $B.attentions$), we obtain attention distribution representations of each sentence for each token from the output results of the BERT model B, as shown in Eq. 3. Assuming that the pre-trained BERT model has L layers with H attention heads per layer, the attention distribution representation of each sentence is $\mathbf{C} \in \mathbb{R}^{L\times H\times N\times N}$, as shown in Eq. 3. Here, N refers to the length of each sentence*[token*$_1$*,...,token*$_N$*]*;

$$\mathbf{C} = B.attentions \tag{3}$$

For each span *[token*$_{l_j}$*,...,token*$_{r_j}$*]* annotated in Sect. 3.1.1, we extract its attention distribution representation $\mathbf{A_{i_j}} = C[:,:,l:r+1,l:r+1]$ from a sentence attention distribution representation $\mathbf{C} \in \mathbb{R}^{L\times H\times N\times N}$.

Candidate Dark Jargon Phrases Selection. Based on the span attention distribution representation $\mathbf{A_{i_j}}$ in Sect. 3.1.2, we need to find a classifier to accurately select the attention distribution representation of the candidate dark jargon phrases P. The span attention distribution representation $\mathbf{A_{i_j}}$ can be viewed as a rectangular picture of pixels of height and width $|l-r| * |l-r|$ with the number of channels $L*H$. Here, $|l-r|$ is the length of phrases. Now we transform the problem of candidate dark jargon phrase selection into an image classification problem, input a span attention distribution representation $\mathbf{A_{i_j}}$, and judge whether the corresponding span *[token*$_{l_j}$*,...,token*$_{r_j}$*]* is a candidate dark jargon phrase P.

Firstly, we adopted a Convolutional Neural Network (CNN) model [9]. Then, we input span attention distribution representation $\mathbf{A_{i_j}}$ to the 3 CNN layers' CNN model f_θ, parameterized by θ, as shown in Eq. 4.

$$p = f_\theta(\mathbf{A_{i_j}}) \tag{4}$$

Here, p is the final classification probability; $\mathbf{A_{i_j}}$ is the span attention distribution representation of the j-th span in i-th sentence.

Secondly, to prevent the CNN model from overfitting, we propose to improve the logistic regression layer by introducing a regularization term in the cost function to penalize the model for having too large parameter values. Then, in the CNN model output layer, we output the Fully Connected layer (FC layer) to the enhanced logistic regression layer. In other words, the FC layer is output to the sigmoid function, to obtain the final classification probability.

Finally, the span attention distribution representation $\mathbf{A_{i_j}}$ is assigned a binary label in the way of threshold division. Here the threshold value is set to θ_p. And the Stochastic Gradient Algorithm (SGD) [2] is used to minimize the value of the cost function on the training set$\{\mathbf{A_{i_j}}, L_{i_j}\}$, as shown in Eq. 5 to continuously optimize the parameter θ, so that the performance of the classification model reaches the optimal.

$$J(\hat{\theta}) = -\min_{\theta} \left(\sum_{i=1}^{n} (L_{i_j} \log p + (1 - L_{i_j}) \log(1 - p)) \right) + \lambda \sum_{j=1}^{m} \theta_j^2 \qquad (5)$$

where L_{i_j} represents the label of the j-th span in i-th sentence; p is the final classification probability of classifier f_θ, parameterized by θ ; J is the cost function; n is the number of the span attention distribution representation $\mathbf{A_{i_j}}$; m is the number of parameters to be learned, and λ is the regularization parameter.

3.2 Phrase-Level Context Representation Generation Module

Based on the fact that the candidate's dark jargon phrases P in Sect. 3.1 have been selected, then we selected the pre-trained BERT model to obtain the high-quality context representation R of these phrases. Unfortunately, at present, most of the English pre-trained BERT models trained on the underground industry corpus D are based on the word level [3].

To generate high-quality phrase-level context representation, We propose the Black-BERT model, a phrase-based pre-trained language model trained on the underground industry corpus D. Firstly, in the Black-BERT model, we add the candidate's dark jargon phrase P to the Tokenizer's vocabulary V of the pre-trained language model to flexibly segment sentences. Secondly, the Tokenizer will loop through the span in each sentence and check whether the span is in the Tokenizer's vocabulary V. If the span is in the Tokenizer's vocabulary V, it will be held on. Otherwise, the native Tokenizer will divide the span. Then, we will concatenate each phrase segmentation result in an ordered way to form the final phrase segmentation result. Finally, Black-BERT continues the mask prediction task based on the "allenai/cs_roberta_base" model.

After fine-tuning the Black-BERT model, we use Tokenizer Black_tokenizer, to tokenize each sentence *[token*$_1$,...,*token*$_N$*]* into phrase segmentation result P=*[phrase*$_1$,...,*phrase*$_M$*]*,e.g. "vendor/review/crack cocaine/on dream". Here, M is the number of the phrase in a sentence.

Then, We map the phrases P=*[phrase*$_1$,...,*phrase*$_M$*]* to their corresponding integer encoding *ids=[id*$_1$,...,*id*$_M$*]*. Next, we format the *ids* as in Sect. 3.1.2. In order to obtain high-quality phrase context representation, inspired by [1], we choose the third-to-last layer of the hidden layer of the Black-BERT Model (e.g. .h_states[10]) as the context representation of phrases, as shown in Eq. 6.

$$\mathbf{R} = Black\_BERT(ids, atten\_mask, out\_h\_states, ret\_dict).h\_states[10] \quad (6)$$

where *atten_mask* and *return_dict* have explained in Eq. 2. *out_h_states* indicates whether to output the hidden states; Finally, We obtain sentence context representation $\mathbf{R}$=*[*$\mathbf{r_1}$,$\mathbf{r_2}$,...,$\mathbf{r_d}$*]* for each full semantic unit, where each $\mathbf{r_i}$ is a 768-dimensional vector and d is the dimension of $\mathbf{R}$.

The context representation of the candidate's dark jargon phrases $\mathbf{r_c}$ is obtained by filtering out the context representation of underground industry terms (a collection of the formal names of illegal products, such as "cocaine"). For convenience, the underground industry terms are simply referred to as terms in the following. In our study, we set three kinds of terms, including drugs, pornography, and weapons. It is next to extract the high-quality contextual representations $\mathbf{r_t}$= $[r_{t,1}, r_{t,2}, ..., r_{t,768}]$ of terms *[T*$_1$,...,*T*$_k$*]*, which is similar to the method of obtaining high-quality phrase context representation above. Here, k is the number of underground industry terms.

3.3 Similarity Calculation's Dark Jargon Phrases Detection Module

The context representations of terms $\mathbf{r_t} = [r_{t,1}, r_{t,2}, ..., r_{t,768}]$ and the context representations of candidate dark jargon phrases $\mathbf{r_c}$= $[r_{c,1}, r_{c,2}, ..., r_{c,768}]$ have been obtained. Then, we select the cosine similarity to compute the similarity *sim* between these two context representations, as shown in Eq. 7.

$$sim = \frac{\mathbf{r_c} \cdot \mathbf{r_t}}{|\mathbf{r_c}| \cdot |\mathbf{r_t}|} = \frac{r_{c,1} * r_{t,1} + r_{c,2} * r_{t,2} + ... + r_{c,768} * r_{t,768}}{\sqrt{r_{c,1}^2 + r_{t,2}^2 + ... + r_{c,768}^2} * \sqrt{r_{t,1}^2 + r_{c,2}^2 + ... + r_{t,768}^2}} \quad (7)$$

If the cosine value *sim* is close to 1, we consider that the direction of these two context representations are the same, and their semantics are similar. Therefore, we need to set the context similarity threshold θ_d. In other words, if the similarity *sim* is greater than the threshold θ_d, the candidate dark jargon phrase will be detected as a dark jargon phrase.

4 Experiments

4.1 Experiment Setting

Dataset. We validate our proposed DJPD model on our dataset related to three underground industries' categories: drugs, pornography, and weapons. In our experiments, we need two inputs: the terms and the underground industry corpus. (1)terms: the term is a collection of official names of illegal products. We refer to [22] for drug terms from the DEA Slang Terms and Code Words list, and pornography and weapons terms from the online slang dictionary. Table 1 summarizes the term's information. (2)the underground industry corpus: We use the original underground industry corpus from social platforms published by Zhu et al. [22]. The data covers three categories: drugs, pornography, and weapons. Table 1 shows the dataset used in our study.

Table 1. Dataset statistics.

Categories	Sentences (numbers)	Underground Industry Terms (numbers)
Drugs	87,876	110
Pornography	81,117	60
Weapons	55,678	130

Evaluation Metrics. Although we conducted our experiments in an unsupervised setting, to better evaluate our experimental results and compare them with state-of-art technologies we use three metrics to evaluate the results, including Recall, Precision, and F1-score [17].

Hyperparameter Settings. In our experiments, the parameters of the model are set as follows.

- Classifier: The batch size of the dataset is 2048, the learning rate is 1e–5.
- Black-BERT model: The parameters to fine-tune this model are as follows: per_device_train_batch_size = 16, learning_rate = 1e–5, mlm_probability = 0.15.
- Threshold: In the experiment, two thresholds of DJPD model in Drugs category to fine-tune are as follows: $\theta_p = 0.19$, $\theta_d = 0.48$.

4.2 Comparison Experiment

Compared Methods. To demonstrate the superiority of our proposed DJPD model in detecting dark jargon phrases, we compare the state-of-the-art methods for dark jargon phrase detection as follows:

- **EPD** [21] uses AutoPhrase [15] to select phrases, then uses Word2vec to filter phrases related to the underground industry, and finally uses SpanBERT's masked language model to obtain the ranking of dark jargon phrases.
- **AutoPhrase** [15] is a data-driven phrase mining tool. We replace the Candidate Dark Jargon Phrase Selection Module with AutoPhrase, and the remaining modules remain unchanged.

- **Word2vec-Skip-gram** [12] is a neural language model. We change the pretrained Black-BERT model in the Phrase-level Context Representation Generation Module to Word2vec-Skip-gram, and the remaining modules of our model remain unchanged.
- **ELMo** [14] is a pre-trained language model which can generate contextual representations of words. Similarly, we replace the Black-BERT model with the ELMo algorithm, and the remaining modules remain unchanged.
- **ChatGPT** [13] is a large language model from OpenAI based on an architecture called GPT-3.5. It can be used to answer questions in a similar way to a human.

Evaluation Results. Table 2 shows the experimental results of the performance of dark jargon phrase detection. It can be seen that our proposed model scores better than other baseline models on our dataset in our study in each metric. It proves the superiority of our proposed DJPD model in detecting dark jargon phrases.

Table 2. Results of comparative experiments. The best results are shown in bold.

Models	Recall(%)	Precision(%)	F1-score(%)
EPD [21]	1.56	1.55	1.55
AutoPhrase [15]	3.12	0.17	0.32
Word2vec+Skip-gram [12]	21.87	34.57	26.79
ELMo [14]	87.02	30.16	44.79
ChatGPT [13]	5.46	6.36	5.88
DJPD(our model)	**87.72**	**84.75**	**86.21**

Firstly, The most robust baseline in our dataset is the ELMo algorithm, which has relatively good performance, but its Precision and F1-score values are much lower (54.59% and 41.42%) compared to the DJPD model. A reasonable explanation is that the ELMO algorithm to generate context representation is based on the word level, without the minimum complete semantic unit phrase as the unit, which will cause semantic ambiguity when generating context representation. In addition, compared with ELMo [14], the performance of Word2vec+Skip-gram [12] is relatively worse. The reason is that Word2vec+Skip-gram can only generate the context representation of static words. Secondly, among the six models, the two with the worst performance are AutoPhrase [15] and EPD [21], which is explained by the fact that the method based on high-frequency n-gram to select candidate dark jargon phrases lacks the ability to detect low-frequency phrases. One thing to point out here is that the ChatGPT [13] has poor performance in jargon phrase detection. It may be because that ChatGPT does not see the jargon phrase before and it can not learn their patterns.

Ablation Experiments. To further understand our proposed DJPD model, we conduct ablation experiments and investigate the effectiveness of several components of our method. Table 3 shows the results of the ablation study.

Ours - noun phrase pseudo-labels, Ours - pre-trained Black-BERT model represent the performance of our model without noun phrase pseudo-labels, and without fine-tuning the BERT model respectively. Compared with our Ours, the performance of Ours-Noun_Phrase_Pseudo-labels and Ours-Pretrain_Black-BERT degrades evidently, with a range of 64.55–66.52% in F1-score. It verifies the effectiveness of the collaborative work of each part. At the same time, it also proves that these two parts are crucial to the detection of dark jargon phrases.

Table 3. Ablation of DJPD model. The best results are shown in bold.

Models	Recall(%)	Precision(%)	F1-score(%)
Ours-Noun_Phrase_Pseudo-labels	24.43	16.49	19.69
Ours-Pretrain_BlackBERT	62.01	13.10	21.66
Ours	**87.72**	**84.75**	**86.21**

5 Conclusions

In this paper, we propose DJPD model, a novel model for low-frequency aware unsupervised dark jargon phrase detection on social platforms. The DJPD model implements a noun phrase tagging method that does not require high-frequency statistics, in conjunction with the pre-trained Black-BERT model that fine-tunes the phrase level. Experimental results have proved that the DJPD model performs significantly better than the most advanced methods on dataset, and it also proves that the proposed detection model is innovative and practical.

Acknowledgements. This work was supported by the Shandong Provincial Key R&D Program of China under Grants No.2021SFGC0401, the TaiShan Scholars Program under Grants No. tsqnz20221146, the Project of Shandong Province Higher Educational Youth Innovation Science and Technology Program under Grant No.2019KJN028, and the Natural Science Foundation of Shandong Province of China under Grants No. ZR2023QF096, and the National Natural Science Foundation of China under Grant No. 61972176.

References

1. Aloraini, A., Poesio, M.: Cross-lingual zero pronoun resolution. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 90–98 (2020)
2. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 421–436. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_25

3. 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)
4. Dryer, M.S.: Noun phrase structure. Lang. Typol. Syntactic Desc. **2**, 151–205 (2007)
5. Jiang, J.A., Nie, P., Brubaker, J.R., Fiesler, C.: A trade-off-centered framework of content moderation. ACM Trans. Comput.-Human Interact. **30**(1), 1–34 (2023)
6. Ke, L., Chen, X., Wang, H.: An unsupervised detection framework for Chinese jargons in the darknet. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 458–466 (2022)
7. Kim, T., Choi, J., Edmiston, D., Lee, S.G.: Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. arXiv preprint arXiv:2002.00737 (2020)
8. Koetsier, J.: Report: Facebook makes 300,000 content moderation mistakes every day. In: Forbes (2020). https://www.forbes.com/sites/johnkoetsier/2020/06/09/300000-facebook-content-moderation-mistakes-daily-report-says/?sh=48a605e254d0
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
10. Li, Y., Cheng, J., Huang, C., Chen, Z., Niu, W.: Nedetector: automatically extracting cybersecurity neologisms from hacker forums. J. Inf. Secur. Appl. **58**, 102784 (2021)
11. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv preprint cs/0205028 (2002)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 1–9 (2013)
13. OpenAI: Gpt-3.5. [Online] (2023). https://openai.com/about
14. Peters, M.E., et al.: Deep contextualized word representations (2018)
15. Shang, J., Liu, J., Jiang, M., Ren, X., Voss, C.R., Han, J.: Automated phrase mining from massive text corpora. IEEE Trans. Knowl. Data Eng. **30**(10), 1825–1837 (2018)
16. Takuro, H., Yuichi, S., Tahara, Y., Ohsuga, A.: Codewords detection in microblogs focusing on differences in word use between two corpora. In: 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), pp. 103–108. IEEE (2020)
17. Wang, H., Hou, Y., Wang, H.: A novel framework of identifying Chinese jargons for telegram underground markets. In: 2021 International Conference on Computer Communications and Networks (ICCCN), pp. 1–9. IEEE (2021)
18. Wang, Y., Su, H., Wu, Y., Wang, H.: SICM: a supervised-based identification and classification model for Chinese jargons using feature adapter enhanced BERT. In: PRICAI 2022, pp. 297–308. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-20865-2_22
19. Yang, H., et al.: How to learn Klingon without a dictionary: detection and measurement of black keywords used by the underground economy. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 751–769. IEEE (2017)
20. Yuan, K., Lu, H., Liao, X., Wang, X.: Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces. In: USENIX Security Symposium, pp. 1027–1041 (2018)
21. Zhu, W., Bhat, S.: Euphemistic phrase detection by masked language model. arXiv preprint arXiv:2109.04666 (2021)
22. Zhu, W., et al.: Self-supervised euphemism detection and identification for content moderation. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 229–246. IEEE (2021)

Multilayer Vision and Language Augmented Transformer for Image Captioning

Qiang Su[1,2] and Zhixin Li[1,2](✉)

[1] Key Lab of Education Blockchain and Intelligent Technology,
Ministry of Education, Guangxi Normal University, Guilin 541004, China
[2] Guangxi Key Lab of Multi-source Information Mining and Security,
Guangxi Normal University, Guilin 541004, China
lizx@gxnu.edu.cn

Abstract. Image Captioning task is one of the important tasks in computer vision. In this paper, we propose a Multilayer Vision and Language Augmented Transformer (MVLAT) method for improving the correctness of image description statements. In MVLAT, we have matched image and text features by adding regional object image features as well as grid image features and introduced them into the image description task by cross-modal retrieval. We refer to this process as the visual contextual relationship module (VRM) to enhance the visual and contextual features of the image description model. In addition, to focus attention more on the focused information, we propose an attention enhancement module (AEM) that enhances the attention weight of important information while weakening the attention weight of non-essential information. Finally, we propose a multi-label Focal loss in the image description task that balances the positive and negative sample imbalance in the training model. Experiments on the MSCOCO image description baseline dataset show that the present method can obtain good performance, and the overall performance of the model is better than many existing state-of-the-art methods. The improvement over the baseline model is 7.7 on the CIDEr score and 1.5 on the BLEU-1 score.

Keywords: Image captioning · computer vision · visual contextual

1 Introduction

A very important task in image description tasks is the rational use of the relationships between the semantic information of the exact target objects extracted by image features. Based on the classical top-down attention mechanism for region feature recognition [1], many state-of-the-art image description tasks [5,6,24] have been studied from two directions: visual relations and positional relations. Nowadays, with the popularity of many Visual Transformer frameworks, which can adequately refine visual and positional relations, many

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 210–222, 2024.
https://doi.org/10.1007/978-981-99-7022-3_19

state-of-the-art models are applied to image description tasks [2,3,8,15,17,22,23] and achieve very excellent results.

Although the previous task has achieved very good results, the modeling process is still very dependent on the quality of the image feature extraction, and the quality of the image description statements is easily affected by the single image feature input vector. As shown in Fig. 1, in the local description, the common description as "people sitting chairs". Obviously, such a description is not comprehensive and does not focus on other important information in the image. By introducing the VRM module, we add visual features and textual features to the original image, which is the visual contextual relationship missing in the original model. With the input of visual contextual relationships, we can add the weight of the object "passengers", the weight of the verb "wait", and the weight of the scene information "at the platform". Diversity of input information helps the model to generate more accurate image captions and, in some cases, can help us to resist interference. Relying only on individual image feature inputs will make it very easy to generate one-sided image captions. If the model is based on diverse feature inputs, it will have rich prior knowledge to assist the model in learning and reasoning about the images. The problem of generating lopsided image descriptions can be greatly alleviated. This is also the starting point of our work, and we can use the rich visual contextual relationships as a priori knowledge to combine into the image captioning generation model to imitate the human reasoning process when seeing images and get more accurate image captions by diversified inputs.

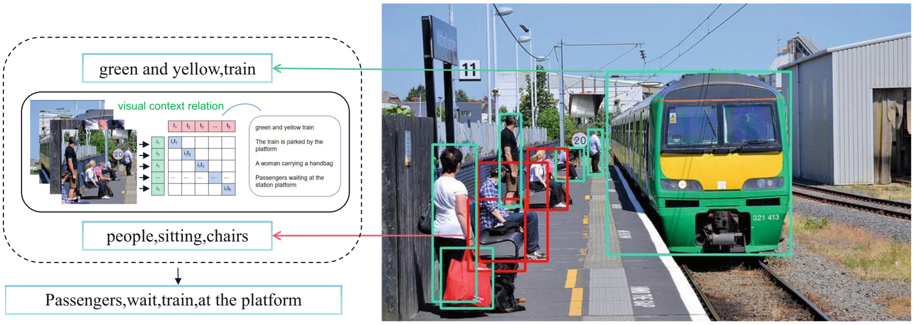

green and yellow,train
visual context relation
t_1 t_2 t_3 ... t_5
i_1 i_2 i_3 ... i_5
i_1t_1 i_2t_2 i_3t_3 ... i_5t_5
green and yellow train
The train is parked by the platform
A woman carrying a handbag
Passengers waiting at the station platform
people,sitting,chairs
Passengers,wait,train,at the platform
11
20
321 413

Fig. 1. Using visual context to discover the implicit meaning in the image and the importance of different words in the image description sentence.

The common methods for image feature extraction include CNN, R-CNN, and faster-RCNN [20]. However, a single feature input tends to generate one-sided image description statements. Therefore, we propose the visual contextual relationship module (VRM), which employs different feature extraction methods to complement the input of visual and textual features. First, visual features in the MSCOCO dataset are extracted by the Visual Transformer Encoder method in CLIP [19]. Second, the text label features in the MSCOCO dataset are extracted by the Text Transformer Encoder method in CLIP. Then, the contextual relationship between image features as well as text features is established by cosine similarity matching, and for an image feature, we take the top k (k = 12) text features with the highest similarity for pairing. Finally, they are merged with the base image features by stacking and fed into the Transformer Decoder to achieve the purpose of compensating the missing visual information of the traditional image description task by additional compensating inputs of visual contexts. We will classify the compensated visual features into three categories: globe, object, and grid. Each of their image regions generates visual contextual relationship pairs that are used to complement the visual contextual relationships in the model. In addition, we propose an attention enhancement module (AEM) to filter the output attention weights. AEM will filter the attention vector weights of some unimportant categories and increase the attention weights of important categories so that our model will focus more on useful information. Since a large number of visual features are added to the model, it causes the problem of positive and negative sample imbalance during the training process. Therefore, we propose a multi-label Focal loss method based on the single-label Focal loss [11] in the training of the model, combined with the original cross-entropy loss function. We use it to balance the weights of positive and negative samples, which makes the training of the model more effective. Finally, by combining VRM, AEM, and the pair-label Focal loss method, the Multilayer Vision and Language Augmented Transformer (MVLAT) method is implemented to form a modeling approach that realizes the combination of image feature information and visual contextual relationship information to achieve the purpose of diversified input.

The primary contributions of this paper are as follows:

- We propose a visual contextual relationship module (VRM) that compensates for the easily missing contextual relationship information of the image description task.
- We propose an attention enhancement module (AEM) that enhances the attentional weight of important information and reduces the attentional weight of easily distracting information.
- We propose a multi-label Focal Loss training method based on Focal loss, which improves the utilization of positive and negative samples in the model and enhances the training of more difficult training samples.
- We propose the MVLAT method, which combines VRM, AEM, and multi-label Focal loss methods. We balance the parameters of the model through a large number of experiments and verify the effectiveness of our method through experiments, and prove that the MVLAT method can make the performance of the model improve effectively.

The other sections of this article are organized as follows: Sect. 2 mainly introduces the method of the model in this article, Sect. 3 presents the experiments and results of our model, and Sect. 4 presents the conclusions of the model.

2 Method

MVLAT is mainly composed of two core modules: VRM and AEM. The overall structure of the model is shown in Fig. 2. The core framework of MVLAT is similar to the classic Transformer model. The model is divided into an encoder-decoder structure. We will mainly introduce these two parts in the following content.

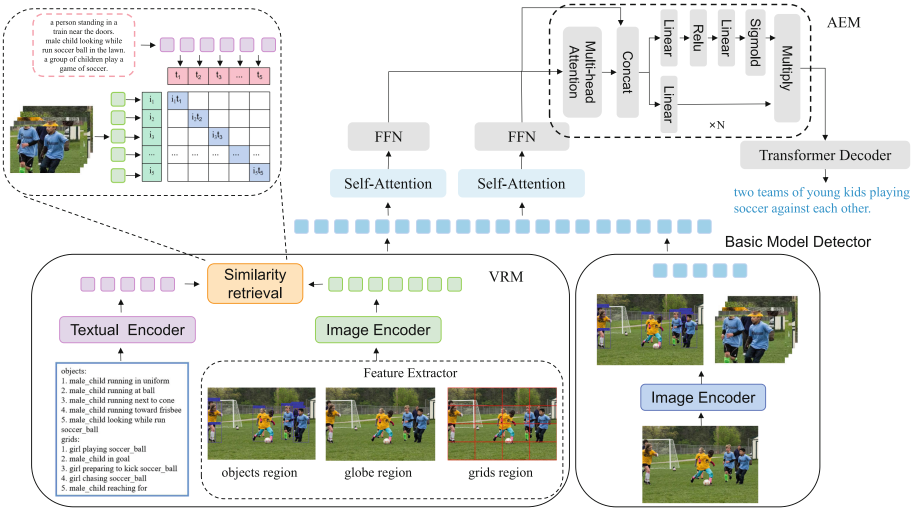

a person standing in a train near the doors.
male child looking while run soccer ball in the lawn.
a group of children play a game of soccer.
t1 t2 t3 ... t5
i1 i2 i3 ... i5
i1t1 i2t2 i3t3 ... i5t5
AEM
Multi-head Attention
Concat
Linear
Relu
Linear
Sigmoid
Linear
Multiply
×N
FFN
FFN
Self-Attention
Self-Attention
Transformer Decoder
two teams of young kids playing soccer against each other.
Basic Model Detector
Similarity retrieval
VRM
Textual Encoder
Image Encoder
objects:
1. male_child running in uniform
2. male_child running at ball
3. male_child running next to cone
4. male_child running toward frisbee
5. male_child looking while run soccer_ball
grids:
1. girl playing soccer_ball
2. male_child in goal
3. girl preparing to kick soccer_ball
4. girl chasing soccer_ball
5. male_child reaching for
Feature Extractor
objects region
globe region
grids region
Image Encoder

Fig. 2. Overall framework of MVLAT.

2.1 Transformer Basic Models

The basic flow formula of a standard Transformer model is as follows:

$$\hat{V} = softmax(\frac{QK^T}{\sqrt{d_k}})V = \Omega_A V, \tag{1}$$

$$MultiHead(Q, K, V) = Concat(H_1, ..., H_h)W^O, \tag{2}$$

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V) = \Omega_A V, \tag{3}$$

$$\mathbf{FFN}(x) = max(0, xW_1 + b_1)W_2 + b_2, \tag{4}$$

where the embedded feature vectors X are used as the input to the first encoder layer, after processed by three learned projection matrices W_Q, W_K and W_V, the attention weights represent the visual relation of different Q and K are computed and the weighted average vectors $\hat{V}$ are obtained according to formula 1, d$_k$ is the constant scaling factor, which is equal to the dimension of W_Q, W_K

and W_V. $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{\frac{d}{h} \times d}$ are head projection matrices for head i, W^O denotes the linear transformation. Where W_1,b_1 and W_2,b_2 are the weights and biases of two fully connected layers.

As shown in Fig. 2, in our MVLAT, image features are also used as the input of self-attention. At the same time, before the input of image feature vectors, the visual context information contained in the preprocessed visual context database is merged into the input image feature vectors to optimize the attention weight matrix as the Q, K matrix-vector of attention. Furthermore, Fig. 3 (a) is the traditional Transformer architecture, and Fig. 3 (b) is the improved Transformer architecture in this paper.

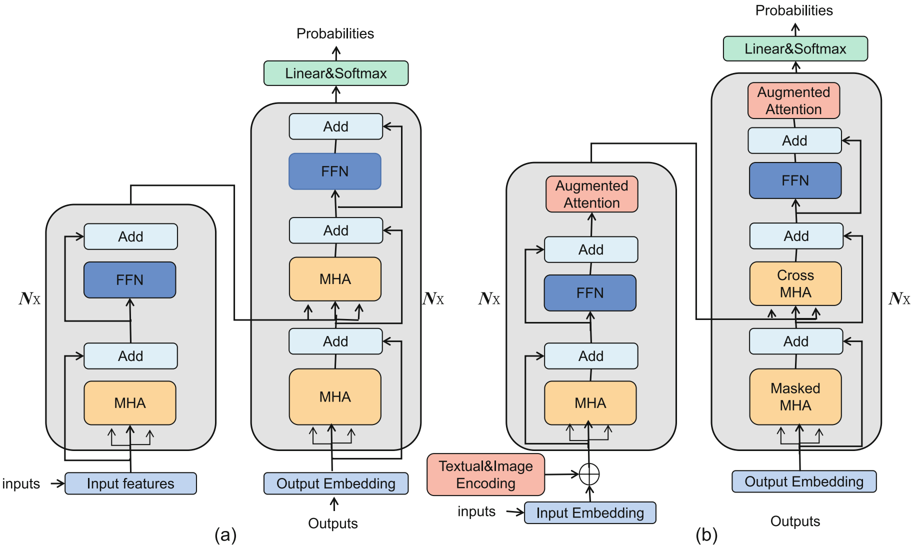


Fig. 3. Comparison between traditional Transformer and our proposed Transformer. Figure (a) shows the traditional Transformer architecture, and Figure (b) shows the improved Transformer architecture in this paper.

2.2 Visual Contextual Relationship Module

In this section, the basic implementation method of the visual contextual relationship module is mainly introduced. The detail process of the visual contextual relationship module shown as the Fig. 4. First of all, this paper establishes a visual context information database containing “attribute object” pairs and “subject predicate object” triples by parsing attributes and relational annotations from Visual Genome. Secondly, based on the ready-made cross-modal joint embedding method in CLIP’s work, we can obtain the image sub-region feature vector and use the image sub-region feature vector to query the relevant text description in the visual context information database.

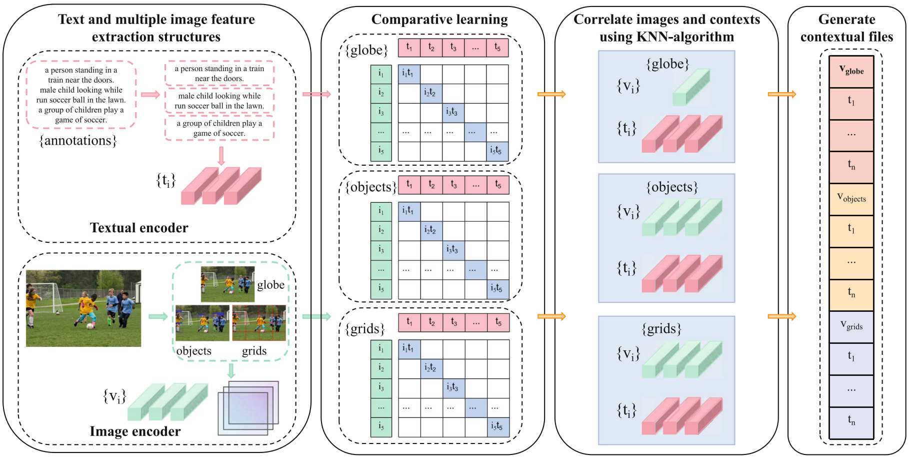

Text and multiple image feature extraction structures
a person standing in a train near the doors.
male child looking while run soccer ball in the lawn.
a group of children play a game of soccer.
{annotations}
{t_i}
Textual encoder
globe
objects
grids
{v_i}
Image encoder
Comparative learning
{globe}
{objects}
{grids}
t_1 t_2 t_3 ... t_5
i_1 i_2 i_3 ... i_5
i_1t_1 i_2t_2 i_3t_3 i_5t_5
Correlate images and contexts using KNN-algorithm
Generate contextual files
v_globe
v_objects
v_grids
t_1
...
t_n

Fig. 4. This is framework of VRM. VRM mainly utilizes CLIP to extract image features of globes, objects and grids.

For the extraction of visual context, we need an image encoder module and a text encoder module. The image encoder module encodes the input image into an image feature vector as visual input. The text encoder module encodes the text annotations under the annotations directory in Visual Genome into a text feature vector, as input to the context of our visual context. In order to maintain the integrity of the model, we use the image encoder and text encoder included in the CLIP model, which is both trained based on the Transformer architecture. In this context, we label them as CLIP-I for the image branch and CLIP-T for the text branch. These branches transform images and text into comprehensive feature representations. CLIP employs a large-scale training approach for image-text pairs, utilizing comparative learning. This method brings pairs of images and text closer together in the embedded space, simultaneously creating separation between unmatched images. Using a pre-trained model, the multi-modal search problem is transformed into a nearest neighbour search problem. The basic processes involved can be summarized as follows:

$$i_f = ImageEncoder(I), t_f = TextEncoder(T), \tag{5}$$

$$i_e = Norm(i_f \cdot W_i), t_e = Norm(t_f \cdot W_t), \tag{6}$$

$$s(i_e, t_e) = \frac{i_e \cdot t_e}{\|i_e\| \times \|t_e\|}, \tag{7}$$

$$i_{vram} = i_{globe} + (1 - \alpha) \times (i_{objects}) + \alpha \times i_{grids}, \tag{8}$$

where I belong to global, objects, grids, indicating the image data that has been divided after object and grid recognition. Images are divided into three types: global, objects, and grids. i_f represents the original global image feature obtained after image encoding. The image encoder can choose to use ResNet or Visual

Transformer. To ensure the integrity of the model, we use the Visual Transformer recommended in CLIP. T comes from annotations, which represents the annotations text label content of Visual Genome. t_f represents the original global text tag feature obtained after passing through a text encoder, where the text encoder can choose RNN or Text Transformer. Like image encoders, to maintain the model's integrity, we use the Text Transformer recommended in CLIP in our article. $W_i$and W_t is the built-in multimodal embedding, i_e and t_e is the enhanced image and text feature vector obtained after multimodal embedding. We use CLIP-T to encode all text annotations in the image description text as search keywords. The main object image sub-region, grid image sub-region, and original image are encoded and queried through CLIP-I. The algorithm of cosine similarity is used to train and calculate the similarity between the two vectors, search for text descriptions with the first k highest cosine similarity scores among text description features, and store them with corresponding image feature vectors for weighted training of subsequent image description tasks. The weight ratio of indexed object regions and grid regions directly affects the final performance of the model, so we set a super parameter α to adjust the proportion of object regions and grid regions in the model. The value of α will be shown in the following experiments.

At this step, we use the CLIP-T that has been pre-trained to encode each set of retrieved text description T into a global representation, obtain the text description vector of visual context information, and then extract the input image feature vector through the pre-trained CLIP-I as the image encoder. Finally, because the model of image caption is typically an auto-regressive model, it is only necessary to concatenate the generated image feature vector and the text description vector containing visual context information to obtain an enhanced image feature input vector with visual context.

2.3 Attention Enhancement Module

The Transformer model is widely used in the self-attention mechanism, which can efficiently extract the internal correlation of features or data. A self-attention mechanism, like most attention mechanisms, needs to obtain three vectors, Q, K, and V, and extract the internal correlation between Q, K, and V through the inner product of Q and K; In the self-attention mechanism, the normalization method is used to maintain the stability of the gradient. The main method is to divide the inner product by the root of the dimension of the K vector, and then obtain a vector that is positively related to attention through the softmax activation function, and then dot the V vector to obtain the final weighted attention vector. The calculation formula for self-attention is as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d}})\mathbf{V}, \tag{9}$$

$$\begin{aligned}\hat{A} = Add\&N(A + \sigma(W_g^q Q + W_g^v f_{mhatt}(Q, K, V) + \\ b^g) \odot (W_i^q Q + W_i^v f_{mhatt}(Q, K, V) + b^i)),\end{aligned} \tag{10}$$

where formula 9 is the standard attention formula. In formula 10, W_g^q, W_g^v, W_i^q and W_i^v is the weight matrix of attention, Q, K, and V are the three input vectors of attention, and f_{mhatt} is the multi-headed self-attention function, and σ is the sigmoid function. In formula 10, the original input feature vector A is invariant. An enhanced attentional feature is obtained by summing the inner product of the upper and lower two channels in the AEM module with the original input features.

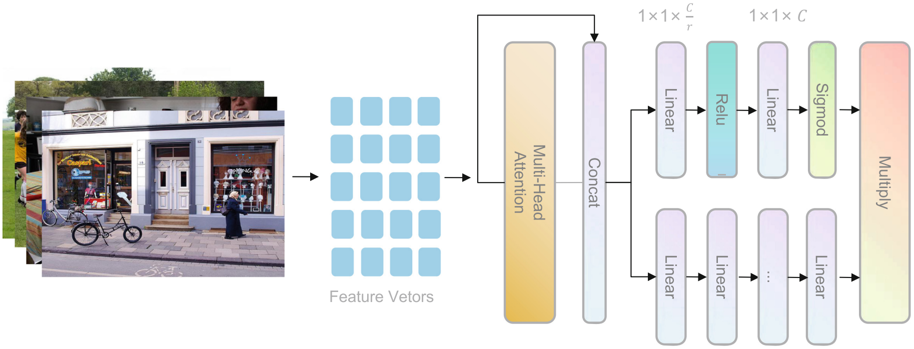


Fig. 5. Framework of the AEM. The function of AEM is divided into two routes, the upper link filters the attention weights by descending and ascending through the operation of the sigmoid, and the lower link keeps the dimensionality unchanged and finally gets the enhanced attention weights by the inner product.

As shown in Fig. 5, combined with the formula, it can be seen that in order to improve the performance of attention usage in Transformer, we have added a compensation attention module on top of the original output attention vector. This compensation attention module constructs the correlation between channels through two fully connected layers, and the number of output weight values is the same as the number of input feature attention vector channels. Through channel descent and ascent operations, finally, the attention vector is more focused on highlighting information through the Sigmod function. Essentially, it is to enable our model to focus more on channel features with large amounts of information while suppressing unimportant channel features, thereby achieving an attention-enhancing effect.

2.4 Objectives

In the training and reasoning stages, the two schemes proposed in this paper can not only be properly combined to guide the description of the generation process, but also each scheme can be implemented separately to solve the different defects in the previous model. The model adopts the mainstream sequence generation method, that is, the description statement is generated from word to word, and the model is usually trained by optimizing the cross-entropy(XE) loss function:

$$L_{XE} = -\sum_{t=1}^{T} log(p_\theta(y_t^* \| y_{1:t-1}^*)), \tag{11}$$

where p_θ is probability and y_t^* is the phrase of the output word.

The multi-label Focal loss based on Focal loss combined with the original cross-entropy loss is given by the following equation:

$$L_{XE}(p, y, \beta_t) = -\beta_t \cdot (1 - p_\theta)^\gamma \cdot \sum_{t=1}^{T} log(p_\theta(y_t^* \| y_{1:t-1}^*)), \tag{12}$$

where β is used as a hyperparameter to regulate the balance between positive and negative samples. When its value is less than 0.5, the model will optimize the training of negative samples. When its value is greater than 0.5, the model will optimize the training of positive samples. Additional, The $(1-p_\theta)^\gamma$ is used as a modulation factor to automatically adjust the training weights of the difficult samples. As the model's probability p_θ approaches 1, the model will decrease the training weight for this classification. As its value approaches 0, the model will increase the training weight for this classification. Differ from the XE loss, we also improve the CIDEr-D score with the beam search. During the decoding phase, we sample the leading k words at every step and retain the highest probability top-k sequences as well:

$$\nabla_\theta L_{RL}(\theta) = -\frac{1}{k} \sum_{i=1}^{k} (r(y_{1:T}^i) - b) \nabla_\theta log p_\theta(y_{1:T}^i), \tag{13}$$

where k is the beam size, r is the CIDEr-D score function, and b=$(\sum_i r(y_{1:T}^i))/k$ is the baseline.

3 Experiments

This section evaluates the effectiveness of the proposed model through experiments. First, it introduces the benchmark dataset and evaluation indicators used in the experiment, and gives details of the implementation of the experiment. Then, it compares the method in this paper with other advanced methods. This paper uses the Karpathy branch noted and uses images to describe the standard evaluation indicators BLEU (B-1, B-4) [18], CIDEr-D (C) [21] , METEOR (M) and SPICE (S) in the task, The results of generating descriptive statements are analyzed quantitatively and qualitatively.

3.1 Implement Details

In this paper, we use DLCT [15], one of the current sota performances, as the baseline model and verify the effectiveness of VRM and AEM through DLCT. For the DLCT baseline model, first use Adam optimizer [9] to train the model under cross-entropy loss. In the beginning, set the learning rate to 1×10^{-4}.

In 10–15 epochs, the learning rate is set to 5 × 10^{-4}. After 15–18 epochs, the learning rate is set to 5 × 10^{-7}. The batch size is set to 100 in the cross-entropy stage. When entering the CIDEr score intensive learning stage, the batch size is set to 50, and the learning rate is set to 5 × 10^{-6}. In the whole process, both the cross entropy training stage and the CIDEr reinforcement learning stage are set to 50. The hyperparameters α and β are selected as shown in Table 1a and 1b, and it referring to the BLEU-1 and CIDEr scores.

Table 1. Performance variation with different α and β values. From this table, we can see that the model can achieve better performance when we set α to 0.6 and set β to 0.8.

α Values	0.4	0.5	0.6	0.7	0.8
BLEU-1	82.9	**83.0**	82.9	82.9	82.8
CIDEr	140.8	141.3	**141.5**	141.4	141.2

(a) α Values

β Values	0.5	0.6	0.7	0.8	0.9
BLEU-1	83.0	**83.1**	83.0	82.9	82.7
CIDEr	141.3	141.2	141.4	**141.5**	141.3

(b) β Values

3.2 Ablation Studies

Since three core modules are proposed in this paper to optimize the model, the main purpose of this section is to make a quantitative analysis of the role of each module on the overall model, and the role generated by each module is shown in Table 2. From the data shown in this Table, either VRM or AEM, or multi-label Focal loss, can more or less improve the performance of the baseline model.

From the ablation experiments in Table 2 , it is recognized that the model-boosted performance is mainly provided by VRM. However, VRM does not perform adequately due to the large amount of interference introduced. Therefore, we propose AEM and multi-label Focal loss to filter most of the noise interference and greatly improve the performance of the model.

Table 2. Performance comparison of baseline model combined with VRM and AEM modules

VRM(image)	VRM(image-text)	AEM	Muti-FLoss	B-1	B-4	M	R-L	CIDEr
–	–	–	–	81.4	39.8	29.5	59.1	133.8
✓	–	–	–	82.0	40.1	29.7	59.3	137.5
–	✓	–	–	82.5	40.8	30.1	59.6	139.8
–	–	✓	–	81.8	39.8	29.6	59.4	135.3
–	–	–	✓	82.1	40.2	29.7	59.5	136.1
–	✓	✓	✓	**82.9**	**41.4**	**30.4**	**59.6**	**141.5**

3.3 Results and Analysis

In order to verify the effectiveness of the MVLAT architecture proposed in this paper, on the basis of ensuring that the evaluation indicators are consistent, we compare the performance results of some mainstream models in recent years and present the final results in the form of tables. We submitted the results generated by the MVLAT model on the MSCOCO online test set to the online server, as shown in Table 3. In addition, the experimental results of MVLAT at the reinforcement learning stage are shown in Table 4.

Table 3. Leaderboard of various methods on the online COCO test server.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METER		R-L		CIDEr	
Model	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
BUTD [1]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
X-Linear [17]	81.3	95.4	66.3	90.0	51.9	81.7	39.9	71.8	29.5	39.0	59.3	74.9	129.3	131.4
AoA-Net [7]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
SGAE [25]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
DLCT [15]	82.0	96.2	66.9	91.0	52.3	83.0	40.2	73.2	29.5	39.1	59.4	74.8	131.0	133.4
M^2-Transformer [3]	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
Ours(MVLAT)	**82.8**	**97.3**	**69.4**	**93.3**	**55.2**	**85.2**	**42.5**	**75.5**	**30.4**	**40.2**	**59.6**	**77.2**	**139.2**	**140.5**

It is clear from Table 4 that after combining VRM and AEM, the CIDEr scores of MVLAT have increased to 141.5%, respectively, by 7.7%. Where the visual features of the base model in our model refer to the pre-processed visual features based on the MSCOCO dataset in Oscar [13] and Vinvl [27]. The performance of MVLAT for most metrics is more competitive than current state-of-the-art models.

Table 4. Performance comparison of VRM and other state-of-the-art models in the reinforcement learning stage.

Models	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr
Up-Down [1]	80.2	36.9	27.6	57.1	117.9
GCN-LSTM [26]	80.5	38.7	28.5	58.5	125.3
AoANet [7]	80.2	38.9	29.2	58.8	129.8
SGAE [25]	80.8	38.4	28.4	58.6	127.8
NG-SAN [4]	80.8	38.8	29.0	58.7	126.3
ORT [5]	80.5	38.6	28.7	58.4	127.8
X-Transformer [17]	80.9	39.7	29.5	59.1	132.8
GET [8]	81.5	38.5	29.3	58.9	131.6
M2 Transformer [3]	80.8	39.1	29.2	58.6	131.2
SCD-Net [14]	81.3	39.4	29.2	59.1	131.6
TransDIC [16]	81.6	39.3	29.2	58.5	132.0
DLCT [15]	81.4	39.8	29.5	59.1	133.8
BPOD(+oscar) [10]	81.5	39.7	30.0	59.5	135.9
OSCAR [13]	–	40.5	–	–	137.6
Ours MVLAT(+oscar)	82.0	40.5	29.6	59.2	136.5
BLIP [12]	–	40.4	–	–	136.7
BPOD(+vinvl) [10]	81.8	41.3	30.1	**59.7**	139.9
VINVL [27]	–	40.9	–	–	140.5
Ours MVLAT(+vinvl)	**82.9**	**41.4**	**30.4**	59.6	**141.5**

4 Conclusions

In this paper, we propose a MVLAT architecture to achieve the purpose of supplementing visual contextual features for the underlying image description model. First, the visual context features are supplemented by VRM. Second, the attention weights are purified by AEM. Then, the positive and negative samples in the model are balanced by multi-label Focal loss. The results show that MVLAT can effectively complement the contextual features of vision and improve the accuracy of image description statements.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Nos. 62276073, 61966004), the Guangxi Natural Science Foundation (No. 2019GXNSFDA245018), the Innovation Project of Guangxi Graduate Education (No. YCSW2023141), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project.

References

1. Anderson, P., He, X., Buehler, C., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR, pp. 6077–6086 (2018)
2. Chen, T., Li, Z., Wu, J., et al.: Improving image captioning with pyramid attention and SC-GAN. Image Vis. Comput. **117**, 104340 (2022)
3. Cornia, M., Stefanini, M., Baraldi, L., et al.: Meshed-memory transformer for image captioning. In: CVPR, pp. 10578–10587 (2020)
4. Guo, L., Liu, J., Zhu, X., et al.: Normalized and geometry-aware self-attention network for image captioning. In: CVPR, pp. 10327–10336 (2020)
5. Herdade, S., Kappeler, A., Boakye, K., et al.: Image captioning: transforming objects into words. In: NeurIPS (2019)
6. Huang, F., Li, Z., Wei, H., Zhang, C., Ma, H.: Boost image captioning with knowledge reasoning. Mach. Learn. **109**(12), 2313–2332 (2020). https://doi.org/10.1007/s10994-020-05919-y
7. Huang, L., Wang, W., Chen, J., et al.: Attention on attention for image captioning. In: ICCV, pp. 4634–4643 (2019)
8. Ji, J., Luo, Y., Sun, X., et al.: Improving image captioning by leveraging intra- and inter-layer global representation in transformer network. In: AAAI, pp. 1655–1663 (2021)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
10. Kuo, C.W., Kira, Z.: Beyond a pre-trained object detector: cross-modal textual and visual context for image captioning. In: CVPR, pp. 17969–17979 (2022)
11. Li, B., Yao, Y., Tan, J., et al.: Equalized focal loss for dense long-tailed object detection. In: CVPR, pp. 6990–6999 (2022)
12. Li, J., Li, D., Xiong, C., et al.: BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML, pp. 12888–12900 (2022)
13. Li, X., et al.: Oscar: object-semantics aligned pre-training for vision-language tasks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 121–137. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_8

14. Luo, J., et al.: Semantic-conditional diffusion networks for image captioning. arXiv preprint arXiv:2212.03099 (2022)
15. Luo, Y., Ji, J., Sun, X., et al.: Dual-level collaborative transformer for image captioning. In: AAAI, pp. 2286–2293 (2021)
16. Mao, Y., Chen, L., Jiang, Z., et al.: Rethinking the reference-based distinctive image captioning. In: ACM MM, pp. 4374–4384 (2022)
17. Pan, Y., Yao, T., Li, Y., et al.: X-linear attention networks for image captioning. In: CVPR, pp. 10971–10980 (2020)
18. Papineni, K., Roukos, S., Ward, T., et al.: BLEU: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
19. Radford, A., Kim, J.W., Hallacy, C., et al.: Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763 (2021)
20. Ren, S., He, K., Girshick, R.B., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99 (2015)
21. Vedantam, R., Zitnick, C.L., Parikh, D.: CIDEr: consensus-based image description evaluation. In: CVPR, pp. 4566–4575 (2015)
22. Wei, J., Li, Z., Zhu, J., et al.: Enhance understanding and reasoning ability for image captioning. Appl. Intell. **53**(3), 2706–2722 (2023)
23. Xian, T., Li, Z., Tang, Z., et al.: Adaptive path selection for dynamic image captioning. IEEE Trans. Circuits Syst. Video Technol. **32**(9), 5762–5775 (2022)
24. Xian, T., Li, Z., Zhang, C., et al.: Dual global enhanced transformer for image captioning. Neural Netw. **148**, 129–141 (2022)
25. Yang, X., Tang, K., Zhang, H., et al.: Auto-encoding scene graphs for image captioning. In: CVPR, pp. 10685–10694 (2019)
26. Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 711–727. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_42
27. Zhang, P., Li, X., Hu, X., et al.: VinVL: revisiting visual representations in vision-language models. In: CVPR, pp. 5579–5588 (2021)

Neural Machine Translation with an Awareness of Semantic Similarity

Jiaxin Li[1], Rize Jin[1](✉), Joon-Young Paik[1], and Tae-Sun Chung[2]

[1] School of Software, Tiangong University, Tianjin, China
jinrize@tiangong.edu.cn
[2] Department of Artifcial Intelligence, Ajou University, Suwon, South Korea

Abstract. Machine translation requires that source and target sentences have identical semantics. Previous neural machine translation (NMT) models have implicitly achieved this requirement using cross-entropy loss. In this paper, we propose a sentence Semantic-aware Machine Translation model (SaMT) which explicitly addresses the issue of semantic similarity between sentences in translation. SaMT integrates a Sentence-Transformer into a Transformer-based encoder-decoder to estimate semantic similarity between source and target sentences. Our model enables translated sentences to maintain the semantics of source sentences, either by using the Sentence-Transformer alone or by including an additional linear layer in the decoder. To achieve high-quality translation, we employ vertical and horizontal feature fusion methods, which capture rich features from sentences during translation. Experimental results showed a BLEU score of 36.41 on the IWSLT2014 *German* → *English* dataset, validating the efficacy of incorporating sentence-level semantic knowledge and using the two orthogonal fusion methods. Our code is available at https://github.com/aaa559/SaMT-master.

Keywords: Sentence Semantic-aware · Multi-branch Attention · Fusion Mechanism · Transformer

1 Introduction

Neural machine translation (NMT) has made significant progress in recent years, thanks to extensive research on deep learning techniques. The encoder-decoder architecture [1–4] has enabled end-to-end training of NMT models. Specifically, the Transformer [3] has led to remarkable improvements in machine translation performance. The attention mechanism in the Transformer model allows the decoder to focus on the most relevant parts of the input sentence, resulting in more coherent sentence generation. In addition, multi-task learning has been employed to improve the generalization ability of NMT models by learning an inductive bias shared between related tasks [5–7]. Furthermore, several NMT models have attempted to achieve state-of-the-art performance by employing

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 223–235, 2024.
https://doi.org/10.1007/978-981-99-7022-3_20

multi-task learning based on the Transformer architecture [14]. However, most of the previous methods have not emphasized enough on the semantic consistency between the input and translated sentences.

Although these methods ensure that the cross-entropy loss used for model training can accurately generate every token in a translated sentence while retaining the semantics of its input sentence, they overlook the fact that an inaccurate generation for even one token can alter the semantics of the entire translated sentence. To address this issue, we propose SaMT, a sentence semantic-aware machine translation model that prioritizes the semantic consistency between input and translated sentences. SaMT is a Transformer-based NMT model that leverages sentence-level semantic knowledge. We train SaMT to extract sentence semantics using Sentence-Transformer [15], which produces vectors that capture the semantics of sentences. We measure the semantic similarity between input and translated sentences with Sentence-Transformer alone or in conjunction with an additional linear layer in the decoder. Our semantic similarity loss is added to the cross-entropy loss for model training. Additionally, we employ vertical and horizontal feature fusion mechanisms to improve the translation quality. These orthogonal feature fusion mechanisms allow our model to capture richer information during translation. Our model was evaluated on the IWSLT-2014 translation task and outperformed the original model [10] by achieving a higher BLEU score of 36.41. This demonstrates the superiority and innovation of our approach.

Our study's main contributions are as follows:

- We propose SaMT, which prioritizes the semantic consistency between input and translated sentences. SaMT combines sentence-level semantic knowledge with a Transformer-based NMT model and explicitly adds the constraint on semantic consistency to the loss function.
- The inter-layer and intra-layer feature fusion mechanisms employed by SaMT improve the model's representational capability due to their orthogonal fusion strategies.
- Our experimental results on the IWSLT2014 *German* → *English* translation task demonstrate the effectiveness of combining sentence-level semantic knowledge with NMT models, as evidenced by a maximum increase of 0.71 BLEU to the original model.

The remainder of this paper is organized as follows: Sect. 2 introduces the related work on NMT. In Sect. 3, we describes the proposed SaMT in detail. Subsequently, we explain how to train our model with the semantic-aware loss function. Our experimental results are presented in Sect. 4. We conclude this paper in Sect. 5.

2 Related Work

Transformer [3], with its encoder-decoder structure and self-attention mechanism, has emerged as the leading architecture for machine translation. Various scaled-up variants of the Transformer model have been extensively explored. However, overly deep Transformer sometimes lead to unexpected performance degradation due to the mere stacking of layers, which causes gradients to vanish or explode. Recent studies [8,9] have addressed the scaling challenge by incorporating inter-layer feature fusion techniques between the encoder and decoder features. Fan et al. [10] introduced a Transformer-based model utilizing multi-branch attention, where the multi-head attention layers of the encoder and decoder form a branch. These studies have demonstrated that fusing and grouping mechanisms can effectively enhance the performance of NMT models.

Cross-entropy loss is widely used as the loss function in machine translation models, and its effectiveness has been demonstrated by the success of numerous neural machine translation models [2,3,11]. However, cross-entropy loss essentially predicts probabilities at the word level. This word-level focus can cause models to primarily perform word-for-word substitution to preserve semantics between input and translated sentences. To encourage higher-level semantic preservation, auxiliary losses have been proposed. Li et al. [12] proposed a hybrid cross-entropy loss to convert translation from a one-to-one to a one-to-many mapping problem. CBBGCA [13] proposed a confidence-based bidirectional global context-aware training framework, which jointly trains the NMT model and the auxiliary conditional mask language model to improve the decoding ability. Unlike CBBGCA, Jung et al. [11] uses various information produced by the decoder to add sentence-level scores to penalize cross-entropy, directly incorporating sentence-level semantics into NMT frameworks. SBERT [16] is a state-of-the-art sentence embedding model in which the semantic vectors of two sentences with similar meanings are located close to each other.

As a summary, it is crucial for NMT models to preserve the semantics of the source and target sentences, rather than simply substituting words. However, the ability to capture semantics is dependent on the data and representations learned by the entire neural network architecture. It is also not feasible to rely solely on cross-entropy loss to ensure sentence-level semantic preservation. To address this, we propose utilizing a well-pretrained, multi-lingual Sentence-Transformer to constrain the training loss, based on the insight that the semantic vectors for two sentences in a translation pair should be similar. Additionally, our model is built upon a sophisticated and scalable Transformer architecture.

3 Sentence Semantic-Aware Machine Translation

This section begins by describing the problem to be addressed, followed by the introduction of a sentence Semantic-aware Machine Translation (SaMT), which is a solution that preserves the meaning of the source sentences in the translated sentences. The architecture of SaMT is then outlined, and subsequently its various modules and techniques are explained in detail.

3.1 Problem Definition

In the field of linguistics, language translation refers to the process of converting a written sentence from a source language into the target language while preserving the meaning of the input sentence. The preservation of semantic consistency is a critical aspect of machine translation. In general, the NMT model is trained by minimizing the difference between input and translated sentences using cross-entropy loss at the word level. The cross-entropy loss function is depicted in Eq. (1).

$$Cross-entropy \quad loss = -\frac{1}{m}\sum_{j=1}^{m}\log p(y_j|y_<j, x) \tag{1}$$

where m is the length of the translated sentence, x is a source sentence, y_j is the ground-truth word (i.e., token) at the j-th position, and $y_{<j}$ is the partial sentence that has been translated before predicting word y_j.

In the realm of machine translation utilizing cross-entropy loss, there exists an underlying assumption that ensuring the accurate prediction of every word in a translated sentence guarantees that the sentence will maintain the same meaning as its source sentence. However, in practice, cross-entropy loss has its limitations in preserving the original meaning. The calculation of cross-entropy loss at the word-level fails to consider the potential impact of incorrectly predicted words on sentence semantics. While an incorrect prediction may only have a minimal effect on the overall loss due to the averaging of the loss over all words in a translated sentence, it may slightly affect or even entirely change the meaning of the input sentence. Therefore, it is imperative to devise a novel loss function that can accurately capture the meaning of a translated sentence as closely as possible to that of its source sentence.

3.2 Model Architecture

We propose the sentence Semantic-aware Machine Translation (SaMT) model, which ensures the consistency between the semantics of the input and translated sentences. The architecture of SaMT is illustrated in Fig. 1, and it comprises three components: an encoder, a decoder, and a semantic vector generator named SVG. Unlike previous works, SaMT employs a pre-trained multi-lingual Sentence-Transformer to construct sentence vector representations for both the input and translated sentences, measures their semantic similarity, and incorporates the differences as an auxiliary loss to the model. Moreover, SaMT utilizes horizontal and vertical fusions to leverage richer features during translation. The subsequent subsections provide detailed explanations of each of these components.

Encoder and Decoder. Despite being extensions of the typical Transformer architecture, the encoder and decoder in the proposed model incorporate the design principles and methods of multi-branch attention Transformer [10] and feature aggregation techniques [8]. This approach enables the model to leverage a more diverse set of information during the translation process, achieved through the use of fusion mechanisms that operate both within and between different layers.

Horizontal Feature Fusion Within Layers. Our model utilizes a basic self-attention module that closely mirrors the standard Transformer. The only difference is that we have replaced all multi-head attention layers with multiple multi-branch attention layers, which we refer to as multi-branch attention layers. As a result, our new module performs a Horizontal Feature Fusion to integrate intra-group information. The improved functionality and flow within the self-attention module of this new structure is illustrated in Fig. 2.

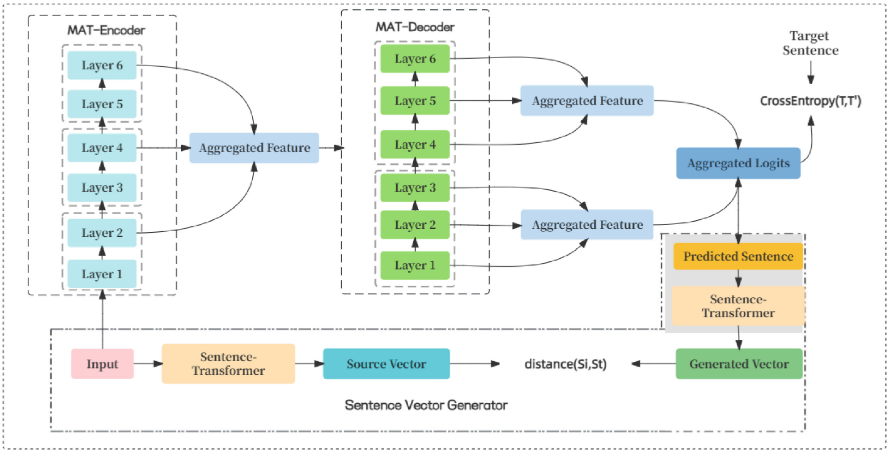


Fig. 1. Architecture of SaMT.(When using the Type II, the shaded areas in the figure will be replaced with SVG_{Linear}, which is a linear layer)

To prevent co-adaptation between branches, we have incorporated the drop-branch technique into our model, where a random selection of branches are dropped during training. Let $MatAttn_{N,M}(Q, K, V; \rho)$ denote a multi-branch attention layer with drop branch rate $\rho \in [0, 1]$. The formula for this technique is expressed as follows:

$$MatAttn_{N,M}(Q, K, V; \rho) = Q + \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbb{I}(U_i \geqq \rho)}{1 - \rho} SelfAttn_M(Q, K, V; \theta_i) \quad (2)$$

where U_i is uniformly sampled from $[0, 1]$, l is the indicator function. $SelfAttn_M$ denotes a multi-head attention layer with M heads. N is the number of branches. $\theta_i = (W_Q^i, W_K^i, W_V^i)_{i=1}^M$ is the parameter set of the i-th multi-head attention layer.

The utilization of a multi-branch attention mechanism by both the encoder and decoder enables our model to extract comprehensive and diverse features.

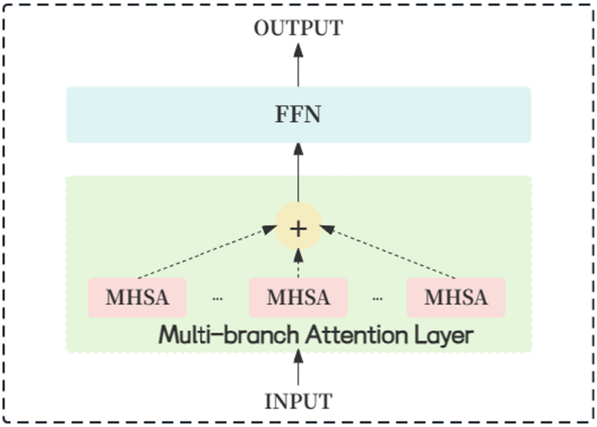


Fig. 2. Architecture of MAT-E Layer

This is because the various branches within the layer focus on different aspects of the same input. The horizontally fused features are integrated within the layers through an average operation, resulting in richer features at every multi-branch attention layer.

Vertical Feature Fusion Between Layers. In order to vertically fuse features from higher to lower layers, we have incorporated the feature aggregation method mentioned in Yang et al. [8]. The fusion of these features, as illustrated in Fig. 1, enables the model to make full use of all layer features, unlike traditional Transformer models that only use the top features. The experimental results have demonstrated that this approach effectively improves the overall performance of the model.

Encoder. All layers of the encoder are grouped to obtain the group features of each group. Different sets of features represent different levels of features. Specifically, let $H_e = h_1^e, ..., h_{L_e}^e$ be the hidden state of the encoder, and L_e represents the number of layers of the encoder. We set $F_e()$ as the fusion function of the encoder. It fuses the hidden state H_e of the encoder into a single representation h_e^f, which is expressed as follows:

$$h_e^f = F_e(H_e) = LN(\frac{1}{N_e}\sum_{i=1}^{N_e}\sigma(w_i^e)h_{\alpha_i}^e) \tag{3}$$

where $N_e = \lceil L_e/T_e \rceil$ is the number of groups of the encoder, and T_e is the number of layers of the encoder in each group. $\alpha_i = min(iT_e, L_e)$; $LN()$ denotes layer normalization; σ is the sigmoid activation function.

Decoder. Let the hidden state of each layer of the decoder be expressed as $H_d = h_1^d, ..., h_{L_d}^d$, where L_d represents the number of decoder layers. Likewise, decoders are also divided into different groups, $N_d = \lceil L_d/T_d \rceil$ is the number of decoder groups, T_d is the number of decoder layers in each group, where the $(k-1)T_d + 1 \sim kT_d$ adjacent layers belong to the k-th group. The k-th fused representation $h_k^{d_f}$ can be computed by describing representation-based merging:

$$h_k^{d_f} = \sum_{i=(k-1)T_d+1}^{min(kT_d, L_d)} \sigma(w_i^{d_r}) h_i^d \tag{4}$$

Subsequently, we get the fused features $[h_1^{d_f}, \cdots, h_k^{d_f}, \cdots, h_{N_d}^{d_f}]$.

3.3 Semantic Vector Generator

Our Semantic Vector Generator (SVG) is designed to produce a unique semantic vector for each sentence in various languages. The purpose of this vector is to facilitate the calculation of semantic similarity between two sentences. This similarity measure is then utilized in the training phase to ensure that translations accurately convey the meaning of the source sentence. We have developed two types of SVGs that are capable of generating semantic vectors for sentences.

Sentence-Transformer Based SVG. The first SVG is based on a pre-trained Sentence-Transformer model, which is a multi-lingual model. This SVG can be shared to generate semantic vectors for both input and translated sentences, which are then located in the same vector space. This approach enables us to leverage the power of pre-trained models and multilingualism to improve the accuracy and efficiency of our translation system.

Equation (5) shows the equation of generating a semantic vector of a source sentence.

$$S^i = SVG_{ST}(I) \tag{5}$$

where SVG_{ST} indicates the SVG based on the pre-trained Sentence-Transformer, $I = (i_1, ..., i_m)$ is an input sentence of m length, and S^i is a semantic vector of the input sentence with d dimensions.

The SVG_{ST} is shared for a translated sentence, as shown in Eq. (6).

$$S^t = SVG_{ST}(T) \tag{6}$$

where $T = (t_1, ..., t_n)$ is a translated sentence of n length, and S^t is a semantic vector of the translated sentence with d dimensions.

It is noteworthy that despite the discrepancy in length between the input and translated sentences, SVG_{ST} is capable of generating vectors of the same dimensionality, facilitating similarity comparisons.

SVG with a Linear Layer. The second SVG, namely SVG_{Linear}, has been proposed to generate the semantic vector of a translated sentence by linearly mapping the decoder's hidden vector. The input sentence, on the other hand, is still generated using SVG_{ST}, Specifically, the output of SVG_{Linear} is a projection of the decoder's final hidden states onto the vector space of the Sentence-Transformer, as depicted in Eq. (7). It is evident that this is a more challenging task, and experimental results have shown that it helps the model achieve better translation performance.

$$S^t = SVG_{Linear}(H) \tag{7}$$

where H is the aggregated vector by the group fusion in the decoder, and S^t is a semantic vector of the translated sentence with d dimensions.

4 Experiments

4.1 Experimental Settings

Cost Functions. As mentioned earlier, we propose a novel cost function to guide our model in explicitly retaining the meaning of source sentences in the translated sentences. This cost function includes the degree of semantic difference between the source and translated sentences in addition to the cross-entropy loss. The proposed cost function can be expressed as follows:

$$L = L_{MT} + \alpha L_{sim} \tag{8}$$

where L_{MT} is the cross-entropy loss, L_{sim} is the semantic similarity distance, and a hyper-parameter α controls the guiding role that the semantic similarity plays in contributing to the total loss. The proposed L_{sim} is inspired by our insight that the semantic vectors of two sentences in the source-target relationship should be located the same or close to each other in the same vector space. The measurement of semantic similarity is the factor that affects the translation quality in our model. We measure L_{sim} by three similarity metrics: cosine similarity, KL divergence, Pearson correlation coefficient. The more similar the meaning two source and translated sentences are, the lower the value of L_{sim} gets. The three metrics are defined as follows:

$$L_{cos} = cos\frac{\sum_{i=1}^{d}(x_i y_i)}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}} \tag{9}$$

$$L_{KL} = KL(softmax(\frac{x_i}{\tau}), softmax(\frac{y_i}{\tau})) \tag{10}$$

$$L_{Pearson} = \frac{\sum_{i=1}^{d}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{d}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{d}(y_i - \overline{y})^2}} \tag{11}$$

On the other hand, L_{MT} is calculated by considering layer groupings. The layers of the decoder are divided into different groups, we take the fused features of all the groups to predict words. For this, we use the probability-based fusion function F_{d_p} to obtain the weighted average probability. Therefore, the multi-level loss is expressed as below:

$$L_{MT} = -\sum_{(x,y\in D)} \sum_{i=1}^{N_d} \psi(w_i^{d_p}) \log P_i(y|x;\theta) \tag{12}$$

where θ represents the model parameters, $\psi(w_i^{d_p})$ is the normalized weight that aggregates the probabilities, and $P_i(y|x;\theta)$ is the translation probability generated by the i-th group of features.

Model Settings. We implemented SaMT with 6 encoder layers and 6 decoder layers. We adopted the default settings provided in the official fairseq as the backbone. The drop-branch rate ρ was 0.3 and the number of branches N was 3. We trained the model using the Adam optimizer and the *inverse_sqrt* learning rate scheduler [3] with an initial learning rate of 5×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.98$. The weight of the final loss function α was 3.

Dataset. We used the IWSLT2014 *German* → *English* dataset for training and testing. The dataset consisted of 16k sentence pairs in the training set and 7k sentence pairs in the validation set.

Model Types. As mentioned earlier, in order to calculate L_{sim}, it is necessary to obtain semantic vectors of two input and translated sentences that have the same dimensions. Therefore, we have designed two combinations of SVG_{ST} and SVG_{Linear} to measure the semantic similarity.

Type I. The SVG_{ST} is shared to generate the semantic vectors of both the source and translated sentences.

Type II. The SVG_{ST} is used to generate the semantic vectors of source sentences while the SVG_{Linear} is used to generate the semantic vectors of translated sentences.

4.2 Results

We conducted an evaluation with two different model types to explore the best approach for implementing SVG, as presented in Table 1. Our proposed method outperforms the original MAT model, achieving a BLEU score of 36.41. This suggests that our method compensates for the decline in semantic understanding caused by using cross-entropy loss by seeking semantic consistency between input sentences and translated sentences. In comparison to other state-of-the-art models, such as Gtrans, which utilizes feature fusion of encoder and decoder on the basis of Transformer to enhance model performance, our two types of models achieve a higher BLEU score by 0.78 and 1.09, respectively. Moreover, we observe a significant improvement in BLEU scores of 1.27 and 1.58, compared to MLRF.

As shown in Table 2, we analyze the performance of the two types under different similarity measures. Both Type I and Type II achieve the best BLEU

Table 1. BLEU-4 scores on the IWSLT-2014 *De* → *En* Task.

Model	BLEU-4
DynamicConv [17]	35.20
MLRF [18]	34.83
ReZero [20]	33.67
Lite-Transformer [19]	33.60
Gtrans [8]	35.32
MAT [10]	**35.70**
SaMT (Type I)	**36.10**
SaMT (Type II)	**36.41**

scores on cosine similarity. The performance of KL divergence and Pearson correlation coefficient is comparable. We believe that cosine similarity can better guide model learning and alleviate semantic understanding problems that may be caused by cross-entropy loss. Comparing the experimental results of the two types under the same measurement method, it can be concluded that Type II is superior to Type I in all measurement indicators. We conclude that the semantic vector obtained by linearly transforming the hidden state of Type II can more directly represent the semantics of translated sentences. Type I is slightly worse, probably because it leads to information loss in the process of generating predicted sentences and semantic vectors, making the semantics of sentences incomplete.

Table 2. BLEU-4 scores of different similarity calculation methods.

Similarity calculation methods	Type I	Type II
Kullback-Leibler	35.78	36.17
Pearson correlation similarity	36.05	36.19
Cosine similarity	**36.10**	**36.41**

Table 3 elaborates the effects of the major components for performance acceleration on performance. The model with the intra-layer and inter-layer feature fusion (i.e., SaMT (w/o similarity)) increases the BLEU score by 0.26. On the other hand, the semantic similarity alone (i.e., SaMT (w/o group)) improves the BLEU scores by 0.09 and 0.24 with Type I and II, respectively. This figure clearly indicates that Type II translates input sentences by capturing their semantics more accurately than Type I. Finally, our SaMT with the two fusion methods and sentence semantics produces the best performance.

Next, we examine the effect of the group size of the encoder on the performance. Type II achieves the best performance when the group size is 3, as shown in Table 4. The performance changes according to the group size. The group size

Table 3. Performance comparsion of different modules.

Model	Type I	Type II
SaMT (w/o both)	35.70	–
SaMT (w/o similarity)	35.96	–
SaMT (w/o group)	35.79	35.94
SaMT	**36.1**	**36.41**

has an effect on the fused feature. When the group size is 1, the features at each layers of the encoder are fused into the aggregated feature without the benefit of the in-group fusion mechanism. Therefore, the aggregated feature is more largely affected by the low-level feature in the group size of 1 than in the larger groups. According to the result, the in-group fusion of the larger groups helps the encoder generate good features, especially with Type II.

Table 4. Performance comparison of different encoder and decoder sizes.

Encoder group size	Type I	Type II
1	36.11	36.08
2	**36.24**	36.15
3	36.10	**36.41**
Decoder group size	Type I	Type II
1	36.07	36.23
2	**36.10**	**36.41**
3	35.99	36.05

Table 4 also shows the effect of the decoder's group size on performance. Type II outperforms Type I in all the three cases of different group sizes. Both types make the best performance when the group size is 2. According to our investigation on this result, larger groups do not lead to the performance enhancement always. Large groups obscure low-level features. It means that the diverse features from high to low levels might help the decoder overcome the lack of the information inherited from the autoregressive property of decoding.

5 Conclusion

In this paper, we present a novel machine translation model, named sentence Semantic-aware Machine Translation (SaMT), to tackle the challenge of semantic similarity between source and target sentences in neural machine translation (NMT). SaMT incorporates Sentence-Transformer into a Transformer-based encoder-decoder architecture to estimate the semantic similarity between sentences. The model utilizes both vertical and horizontal feature fusion techniques to capture rich features from sentences during the translation process. Experimental results on the IWSLT2014 *German* → *English* dataset demonstrate that SaMT outperforms competing models, achieving a BLEU score of 36.41. The proposed model explicitly imposes a constraint on semantic consistency in the loss function, which constitutes a significant contribution to NMT research. In future work, we plan to evaluate the generalization capability of SaMT for translations between diverse languages.

Acknowledgment. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-0-02051) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

1. Sutskever, I., Vinyals, O., Le, Q. V.: Sequence to sequence learning with neural networks. In: Proceedings of the Neural Information Processing Systems, NIPS, pp. 3104–3112. MIT Press (2014). https://nips.cc/Conferences
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of International Conference on Learning Representations, ICLR (2015)
3. Vaswani, A., et al.: Attention is all you need. In: Proceedings of Neural Information Processing Systems, NIPS, pp. 5998–6008. MIT Press (2017). https://nips.cc/Conferences
4. Song, L., Gildea, D., Zhang, Y., Wang, Z., Su, J.: Semantic neural machine translation using AMR. Trans. ACL **7**(1), 19–31 (2019)
5. Zhou, S., Zeng, X., Zhou, Y., et al.: Improving robustness of neural machine translation with multi-task learning. In: Proceedings of the Fourth Conference on Machine Translation, pp. 565–571 (2019)
6. Sánchez-Cartagena, V.M., Esplá-Gomis, M., Pérez-Ortiz, J.A., et al.: Rethinking data augmentation for low-resource neural machine translation: A multi-task learning approach. In: Proceedings of EMNLP, pp. 15–26 (2021). https://2021.emnlp.org/
7. Domhan, T., Hieber, F.: Using target-side monolingual data for neural machine translation through multi-task learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1500–1505 (2017)
8. Yang, J., Yin, Y., Yang, L., et al.: GTrans: grouping and fusing transformer layers for neural machine translation. IEEE/ACM Trans. Audio Speech Lang. Process. **31**, 1489–1498 (2022)
9. Liu, X., Duh, K., Liu, L., Gao, J.: Very deep transformers for neural machine translation. arXiv preprint arXiv:2008.07772 (2020)
10. Fan, Y., Xie, S., Xia, Y., et al.: Multi-branch attentive transformer. arXiv preprint arXiv:2006.10270 (2020)
11. Jung, H., Kim, K., Shin, J.H., Na, S.H., Jung, S., Woo, S.: Impact of sentence representation matching in neural machine translation. Appl. Sci. **12**(3), 1313 (2022)
12. Li, H., Lu, W.: Mixed cross entropy loss for neural machine translation. In: Proceedings of ICML, pp. 6425–6436. ACM (2021)

13. Zhou, C., Meng, F., Zhou, J., Zhang, M., Wang, H., Su, J.: Confidence based bidirectional global context aware training framework for neural machine translation. In: Proceedings of ACL, Dublin, pp. 2878–2889. Association for Computational Linguistics, Stroudsburg (2022)
14. Luong, M.T., Pham, H., Manning, C.D.: Bilingual word representations with monolingual quality in mind. In: Proceedings of VS@HLT-NAACL, pp. 151–159. Association for Computational Linguistics, Stroudsburg (2015)
15. Pretrained Models: Sentence-Transformers documentation. https://www.sbert.net/docs/. Accessed 2023
16. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
17. Wu, F., Fan, A., Baevski, A., et al.: Pay less attention with lightweight and dynamic convolutions. arXiv preprint arXiv:1901.10430 (2019)
18. Wang, Q., Li, F., Xiao, T., et al.: Multi-layer representation fusion for neural machine translation. arXiv preprint arXiv:2002.06714 (2020)
19. Wu, Z., Liu, Z., Lin, J., et al.: Lite transformer with long-short range attention. arXiv preprint arXiv:2004.11886 (2020)
20. Bachlechner, T., Majumder, B.P., Mao, H., Cottrell, G., McAuley, J.: ReZero is all you need: fast convergence at large depth. In: Proceedings of Uncertainty in Artificial Intelligence, pp. 1352–1361. PMLR (2020)

Optimizing Answer Representation Using Metric Learning for Efficient Short Answer Scoring

Bo Wang[1](✉), Billy Dawton[1], Tsunenori Ishioka[2], and Tsunenori Mine[1]

[1] Department of Information Science and Technology, Kyushu University, Fukuoka, Japan
wang.bo@m.ait.kyushu-u.ac.jp, mine@ait.kyushu-u.ac.jp

[2] The National Center for University Entrance Examinations, Tokyo, Japan

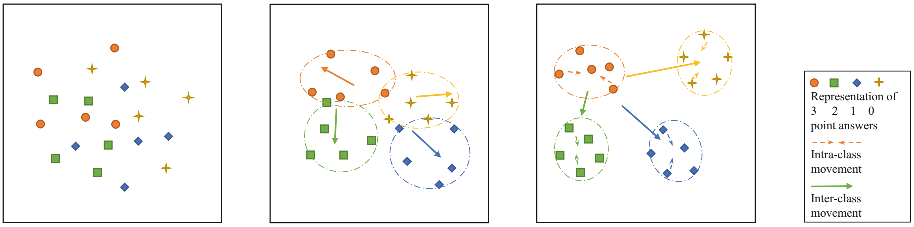
Abstract. Automatic short answer scoring (ASAS) has received considerable attention in the field of education. However, existing methods typically treat ASAS as a standard text classification problem, following conventional pre-training or fine-tuning procedures. These approaches often generate embedding spaces that lack clear boundaries, resulting in overlapping representations for answers of different scores. To address this issue, we introduce a novel metric learning (MeL)-based pre-training method for answer representation optimization. This strategy encourages the clustering of similar representations while pushing dissimilar ones apart, thereby facilitating the formation of a more coherent same-score and distinct different-score answer embedding space. To fully exploit the potential of MeL, we define two types of answer similarities based on scores and rubrics, providing accurate supervised signals for improved training. Extensive experiments on thirteen short answer questions show that our method, even when paired with a simple linear model for downstream scoring, significantly outperforms prior ASAS methods in both scoring accuracy and efficiency.

1 Introduction

Automatic short answer scoring (ASAS) is an important natural language processing (NLP) application used in education, which not only lightens the teacher workload but also addresses the problem of evaluation inconsistency.

A variety of approaches for performing ASAS have been proposed. In early times, short answer scoring was treated as a machine learning problem where discrete features, such as *answer length* [10] or *TF-IDF* [2], were extracted manually and then classified using support vector machines or decision trees. Later, with the emergence of deep neural networks, it became common to let the answer texts pass through a recurrent neural network (RNN) and perform a supervised learning for score prediction [1]. More recently, the advent of large-scale Transformer encoders, especially Bidirectional Encoder Representations from Transformers (BERT [5]), have extended new state-of-the-art results in many tasks including ASAS by providing more comprehensive, contextual embeddings. The

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 236–248, 2024.
https://doi.org/10.1007/978-981-99-7022-3_21



(a) w/o training

(b) applying pre-training or fine-tuning

(c) applying metric learning

Fig. 1. Diagram of answer representation optimization effects using past methods and metric learning

use of BERT-based ASAS methods can be mainly divided into three groups: (i) using a pre-trained BERT as a word embedding encoder and feeding the embeddings into RNNs [24], (ii) fine-tuning a BERT directly [12], or (iii) performing a second-stage BERT pre-training before the downstream scoring process [18].

However, while employing more advanced encoders, past ASAS methods have predominantly relied on self-supervised pre-training (such as masked language model target in BERT training) or supervised training using Cross-Entropy (CE) or Mean Squared Error (MSE) loss to adapt the encoder for scoring. The fundamental steps in applying ASAS have generally mirrored those used in traditional text classification/regression tasks. As shown in recent work [9,15], although CE/MSE supervised training focuses on giving correct classifications, they are still weak in forming label-distinct clusters, giving loose boundaries between them (Fig. 1-(b), let alone the unsupervised methods). As a result, there are considerable overlaps among the representations of the answers with different scores, which may lead the classifier to misclassify more frequently and in turn, impact the overall scoring accuracy.

To address this issue and achieve better accuracy, we propose to use metric learning (MeL [22]) to optimize the answer representations. Originally proposed and widely applied in computer vision fields, MeL operates on a simple yet effective principle: it brings the similar representations together while pushing dissimilar ones further apart by the given metric. By employing MeL, we can effectively form compact embedding clusters for answers at each score level, while pushing clusters of different scores further apart and enhancing their separation (Fig. 1-(c)). Consequently, the differences in their representations are better exposed, benefiting the downstream scoring procedure.

To exploit the full potential of MeL, another important thing to consider is what "metric" should we use. An intuitive choice is the label-based metric, in which, for a given answer (referred to as the "anchor" example), answers with different labels (scores) are treated as the "negative" examples, and those with the same score are considered as the "positive" examples. However, we argue that answers often have a nuanced but stronger similarity relationship, particularly when addressing the same question, which can not be fully expressed by mere score labels. To quantify and leverage this relationship, while considering data availability, we introduce two novel similarity definitions as the metric for MeL: a score-based similarity for the most common scenarios, and a more detailed and precise rubric-based similarity when scoring criteria are accessible.

By conducting MeL pre-training, finally, a linear network is sufficient to deliver superior scoring results. In summary, our main contributions are as follows: We propose an efficient MeL-based method to create a distinctly separated answer embedding space, enhancing scoring efficiency; we innovatively consider two answer similarity definitions based on scores and question rubrics; extensive experiments on thirteen questions across two languages demonstrate the superiority of our method to existing scoring methods in both accuracy and efficiency.

2 Related Work

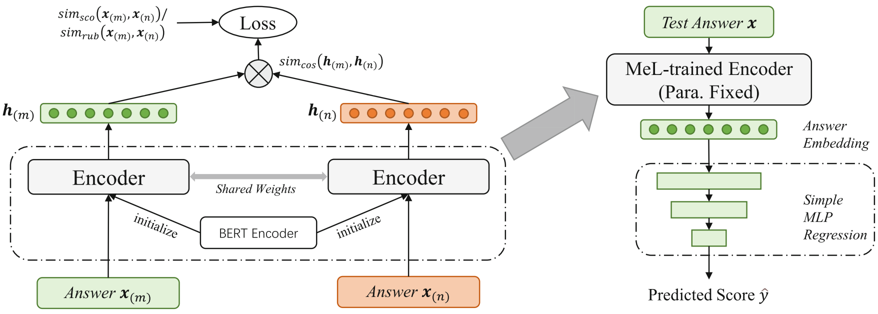
The development of automatic scoring systems parallels the progression of text feature extraction techniques: In early research, handcraft discrete features such as *answer length* [10], *n-grams* [16], and *TF-IDF* [2] were used for scoring. Despite offering better interpretability, they require extensive feature design and network tuning to obtain satisfactory results. After the emergence of deep neural networks, feature extraction has evolved into an automatic process, where employing static word embeddings (e.g., word2vec, GloVe) with a variety of network structures, such as Long Short-Term Memory (LSTM) [1], hierarchical Convolution Neural Networks (CNN)/CNN [6], and CNN/LSTM with attention [7], became the most common implementations. More recently, since the advent of large-scale pre-trained language encoders such as BERT, using them to provide a contextual and comprehensive text representation for the answers has become commonplace. Approaches typically involved fine-tuning BERT directly [12,14,23] or performing a second pre-training iteration before scoring [18,21]. Some recent studies have also adopted strategies that either combine BERT as the encoder with RNNs as the scoring network [13,19,24] or integrate features extracted from BERT along with hand-crafted features [11] for final scoring.

From the review above, it is clear that while there have been advancements in answer feature extraction techniques, existing research rarely focuses on improving encoder training to produce superior answer representations. To the best of our knowledge, our work is the first to utilize metric learning in conjunction with score and rubric information to optimize the formation of answer representations. Several papers have claimed "to consider rubrics", but they mostly only added the rubric contents to the answer inputs [3,4], or additionally applied laborious word embedding re-weighting before scoring [20]. We will show later that this initial usage cannot outperform ours in Sect. 5.

3 Methodology

3.1 Problem Definition

Task: The task of ASAS can be formulated as a supervised machine learning problem, where the objective is to learn a mapping from a short answer $\boldsymbol{x}$ to its corresponding score y, i.e., f: $\boldsymbol{X} \rightarrow Y$, given the training data $(\boldsymbol{x}_i, y_i)_{i=1}^{N}$.


$sim_{sco}(\boldsymbol{x}_{(m)},\boldsymbol{x}_{(n)})/$
$sim_{rub}(\boldsymbol{x}_{(m)},\boldsymbol{x}_{(n)})$
Loss
$sim_{cos}(\boldsymbol{h}_{(m)},\boldsymbol{h}_{(n)})$
$\boldsymbol{h}_{(m)}$
$\boldsymbol{h}_{(n)}$
Encoder
Shared Weights
Encoder
initialize
BERT Encoder
initialize
Answer $\boldsymbol{x}_{(m)}$
Answer $\boldsymbol{x}_{(n)}$
Test Answer $\boldsymbol{x}$
MeL-trained Encoder (Para. Fixed)
Answer Embedding
Simple MLP Regression
Predicted Score $\hat{y}$

(a) Metric learning procedure

(b) Downstream scoring

Fig. 2. Proposed similarity-MeL pre-training – MLP scoring framework

Workflow: A common ASAS workflow first extracts the feature (representation) $\boldsymbol{h}$ of the answer $\boldsymbol{x}$, then predicts the score $\hat{y}$ based on it. This work used to be performed manually, but is now typically done by feeding the answer text to a fixed word embedding layer or directly to a language encoder:

$$answer\ \boldsymbol{x} \xrightarrow{encode} embedding\ \boldsymbol{h} \xrightarrow{scoring} score\ \hat{y}$$

On this point, however, past methods often overlook the upstream training process (i.e., $\boldsymbol{x} \rightarrow \boldsymbol{h}$), but mainly focus on adding more contents to the input $\boldsymbol{x}$, switching to heavier encoders, or applying more complex downstream scoring methods. In this paper, we employ MeL to refine the encoder training process to produce better answer representations, thus enable achieving superior scoring results using only a simple feed forward network (Fig. 2).

3.2 Preliminary: Metric Learning

Metric learning (MeL) is a well-established method widely applied in both the field of computer vision [9] and natural language processing tasks [17]. It employs a simple but effective approach that strives to reduce the distance between similar examples while simultaneously increase the distance between dissimilar ones in the representation space. Compared to traditional machine learning techniques, MeL is able to provide a new data representation space that offers clearer and more meaningful discrimination between categories. Given these advantages, we believe applying MeL to ASAS should also give better representations for the answers and thus improve final scoring.

There are two ways to apply MeL. A typical and convenient way is to designate certain anchor-positive/negative example pairs for learning by the label. For instance, if we use this mode for ASAS encoder training, for a specific answer (the anchor text), we can assign the answers with different scores as its negative examples, and those with the same score as the positive examples. When conducting MeL, the anchor-positive examples are brought closer together and the negatives are pushed further apart. This is realized through a Triplet loss [8]:

$$Loss_{MeL-tri} = max(||\boldsymbol{h}_{anc} - \boldsymbol{h}_{pos}|| - ||\boldsymbol{h}_{anc} - \boldsymbol{h}_{neg}|| + \varepsilon,\ 0) \tag{1}$$

However, the drawback of this learning mode is obvious, as it treats answers with different scores as isolated categories. We propose that answers to the same question should exhibit stronger similarity relationships – a subtle concept that cannot be entirely captured by score labels alone. Fortunately, there is another learning mode of MeL that can exactly leverage these similarity relationships between examples. Specifically, in this mode, we need to construct example pairs as follows: *[answer 1, answer 2, sim(ans1, ans2)]*, and implement MeL in a Siamese structure (Fig. 2-(a)) with the following mean squared error loss:

$$Loss_{MeL-sim} = [sim(\boldsymbol{h}_{ans1}, \boldsymbol{h}_{ans2}) - sim_{gt}(\boldsymbol{x}_{ans1}, \boldsymbol{x}_{ans2})]^2 \tag{2}$$

where sim_{gt} denotes the defined, $\underline{g}$round $\underline{t}$ruth similarity between two answers, while the embedding similarity is usually calculated using the cosine distance:

$$sim_{cos}(\boldsymbol{h}_{ans1}, \boldsymbol{h}_{ans2}) = \frac{\boldsymbol{h}_{ans1} \cdot \boldsymbol{h}_{ans2}}{||\boldsymbol{h}_{ans1}||||\boldsymbol{h}_{ans2}||} \tag{3}$$

It is clear that employing this method provides more precise supervision signals for MeL, thereby enhancing the efficiency of the encoder's training process.

3.3 Defining the Answer Similarity with Score and Rubric

Having determined the learning mode of MeL, we next need to consider a reasonable definition of similarity between the answers since it is not explicitly included in the original training data $(\boldsymbol{x}_i, y_i)_{i=1}^{N}$. We propose two definitions suited to different application scenarios and data availability. Note that to prevent confusion, in the rest of the paper, we use symbols like $\boldsymbol{x}_{(m)}$ to denote answers $\boldsymbol{x}$ that receive an m-point score.

1) Score-based Similarity

In situations where only $(answer, score)$ data is available, we propose to simply use the score ratio as a quick estimate of the answer similarity:

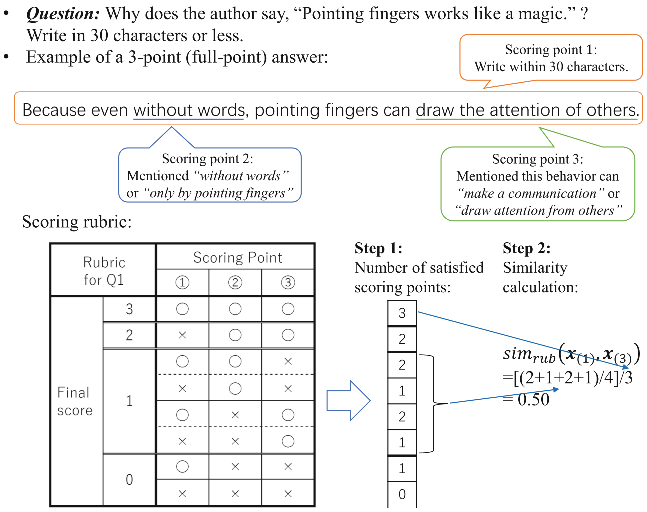
Definition 1 (score-based similarity). *Let $\boldsymbol{x}_{(m)}$ and $\boldsymbol{x}_{(n)}$ represent answers with scores of m and n points. The similarity between them is defined as:*

$$sim_{sco}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) = \frac{m}{n}\ (m \leq n) \tag{4}$$

2) Rubric-Based Similarity

While the score-based similarity is straightforward and easy to compute, it provides only a basic estimation and may not fully reflect the intrinsic relationships between answers. We propose to refine this by further incorporating "rubric" data into the similarity definition.

The inspiration comes from the actual scoring process. Answers are generally evaluated in several separated sections, each corresponding to a "scoring point". Therefore, we can use the comparison of fulfilled scoring points between two



- ***Question:*** Why does the author say, "Pointing fingers works like a magic." ? Write in 30 characters or less.
- Example of a 3-point (full-point) answer:

Because even without words, pointing fingers can draw the attention of others.

Scoring point 1:
Write within 30 characters.
Scoring point 2:
Mentioned *"without words"* or *"only by pointing fingers"*
Scoring point 3:
Mentioned this behavior can *"make a communication"* or *"draw attention from others"*

Scoring rubric:

Rubric for Q1		Scoring Point ①	②	③
Final score	3	○	○	○
	2	×	○	○
	1	○	○	×
		×	○	×
		○	×	○
		×	×	○
	0	○	×	×
		×	×	×

Step 1:
Number of satisfied scoring points:
3
2
2
1
2
1
1
0
Step 2:
Similarity calculation:
$sim_{rub}(\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(3)})$
=[(2+1+2+1)/4]/3
= 0.50

Fig. 3. An example question, its model answer, rubric and the rubric-based similarity calculation process

answers as an approximation for their similarity, where these scoring point rules and final scoring criteria are indeed contained within the scoring rubrics.

For clarity, we here give a specific example of the answer evaluation process using rubric in Fig. 3. The calculation of rubric-based similarity is also shown there: using the rubric "reversely", we determine the average number of scoring points that an answer must meet to achieve a certain score, and then calculate the similarity as the ratio of those fulfilled numbers between two answers.

Definition 2 (rubric-based similarity). *Let $R_{(m)}$ and $R_{(n)}$ be the average number of satisfied scoring points of $\boldsymbol{x}_{(m)}$ and $\boldsymbol{x}_{(n)}$. The rubric-based similarity is defined as:*

$$sim_{rub}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) = \frac{R_{(m)}}{R_{(n)}} \quad (m \leq n) \tag{5}$$

where $R_{(m)}$ (and $R_{(n)}$ in the same way) is calculated as:

$$R_{(m)} = \frac{1}{k_{(m)}} \sum_{i=1}^{k_{(m)}} sp_{(m),i} \tag{6}$$

where $sp_{(m),i}$ is the number of satisfied scoring points for the i-th possible answer case with score m, and $k_{(m)}$ is the total number of such possible cases.

3) Further Refinements to sim_{sco} and sim_{rub}

Considering the fact that answers with the same scores still have minor semantic differences, and the potential data imbalances, we propose two further refinements to the definitions of score-based and rubric-based answer similarities.

(a) The first kind of refinement simply subtracts five percentage points from the defined similarity sim_{sco} or sim_{rub}:

$$sim^{(s)}_{sco}/sim^{(s)}_{rub}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) = sim_{sco/rub}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) - 0.05 \tag{7}$$

Table 1. Basic information of EN-SAS

Question e.g.	Score range	Word limit	Num for pre-training	Num used in MeL	Num for scoring
sci-Q1	0–3	50	1,672	*1,356*	558
Eng-Q3	0–2	50	1,808	*1,450*	406
bio-Q5	0–3	60	1,795	*374*	599

Table 2. Basic information of JP-SAS

Question No.	Score range	Char. limit	Num for pre-training	Num used in MeL	Num for scoring
Q1	0–3	30	id. 1–10k (10,000)	*7,138*	id. 10–20k (10,000)
Q2		40		*8,260*	
Q3		80–120		*6,784*	

(b) The second kind of amendment is more complex, considering possible data imbalances and the variety of answer cases, and is used exclusively for sim_{rub}:

$$sim^{(c)}_{rub}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) = sim_{rub}(\boldsymbol{x}_{(m)}, \boldsymbol{x}_{(n)}) - [\alpha(r(m)+r(n)) + \beta(v(m)+v(n))] \quad (8)$$

where $r(m)$ (and $r(n)$) denotes the ratio of the number of used m-point answers to the total available data, and $v(m)$ (and $v(n)$) denotes the answer variety as the entropy of possible answer cases obtaining m points:

$$r(m) = \frac{|\boldsymbol{x}^{\text{used}}_{(m)}|}{|\boldsymbol{x}^{\text{total}}|} \qquad v(m) = -log_2 \frac{1}{k_{(m)}}$$

To fit the range of cosine similarity of [0,1], we set α=β=0.1 empirically this time, and leave systematic parameter optimization for future work.

3.4 Downstream Scoring Procedure

Upon being trained using MeL with defined similarities, the encoder exhibits an enhanced ability to recognize semantic differences among answers of varying scores, thereby producing more distinguishable representations for scoring. As a result, a simple feed-forward network structure, such as a Multi-Layer Perceptron (MLP), can perform the downstream scoring process efficiently and accurately (Fig. 2-b), eliminating the need for time-intensive RNN-based training or full-parameter BERT fine-tuning. Throughout this paper, we refer to our proposed method as a whole as ***sim-MeL-mlp***.

4 Experiments

4.1 Datasets

We evaluate our method on diverse datasets comprising thirteen questions: the first ten are from the English ASAP-SAS project across three subjects, and the remaining are from Japanese Common University Entrance Examination trial test, 2018. Detailed dataset compositions can be found in Tables 1 and 2.

EN-SAS Dataset: Provided by the ASAP-SAS project (https://kaggle.com/c/asap-sas), this dataset contains ten questions spanning science, biology, and English reading. The given rubrics are generally simplistic and somewhat vague (e.g., *"3 points: Three key elements."* or *"The response demonstrates an exploration of the ideas in the text."*), resulting in minimal differences between sim_{sco} and sim_{rub}. Therefore, we solely apply sim_{sco} for MeL on this dataset.

JP-SAS Dataset: This internal dataset, sourced from Japan National Center for University Entrance Examinations, consists of three Japanese reading questions. The provided rubrics are more explicit and detailed (e.g., the rubric shown in Fig. 3 is the exact rubric of Q1 of this dataset), enabling us to apply both sim_{sco} and sim_{rub} with MeL for evaluation on this dataset.

4.2 Metrics

In our experiments, we frame the scoring task as a regression problem, using mean absolute error (MAE) and root mean square error (RMSE) to compare model performances. Lower values of both metrics indicate better performance.

4.3 Baselines

We selected the following representative ASAS methods not requiring additional pre-processing or handcraft feature generation as the baseline methods.

(1) ***bert-pre-lstm*** [19,24]: This method pre-trains BERT twice using the Masked Language Modeling (MLM) target, with the answer texts from all questions serving as the pre-training data. The generated word embeddings are then processed by a BiLSTM network with attention to predict the score.
(2) ***bert-pre-fine*** [18,21]: This method also performs a twice pre-training on BERT, but then fine-tunes pre-trained BERT directly on the scoring data.
(3) ***bert-fine-fine*** [12,23]: Despite having more texts, the *bert-pre*-based baselines did not use score labels during pre-training, thus may give inferior results to the proposed method. To offer a fair comparison, we apply this method which performs supervised fine-tuning on the data for pre-training, and performs a twice fine-tuning on the scoring data.
(4) ***(rub text)-fine-fine*** [3,4]: A method similar to *bert-fine-fine*, but adds the corresponding rubric text to the answer text as input during the first-stage fine-tuning.
(5) ***label-MeL-mlp***: An ablation study that uses the label as the metric to perform MeL (i.e., Triplet loss as described in Sect. 3.2 and Eq. 1).
(6) ***half/fourth-data***: An efficiency study on JP-SAS dataset, using only half or a quarter of the data to build answer pairs for *MeL-mlp*.

4.4 Implementation Details

Data usage: the training set of both datasets are served as pre-training data for all models, with scoring performance assessed through 5-fold cross-validation on the test set. Encoders: we used “bert-base-uncased” and “bert-large-wwm” for EN-SAS, and “bert-jp-base/-base-wwm” for JP-SAS. Scoring networks: A BiLSTM network with 3 layers (300 neurons/layer) and attention enabled was used for *bert-pre-lstm*. For our methods, a 5-layer MLP network (512,256,128,32,1 neurons/layer) sufficed. Loss: Mean Squared Error was used as the loss function for scoring. Infrastructure: Experiments were run on a machine with Ubuntu 18.04, 128 GB RAM, an Nvidia RTX8000 GPU, and an Intel i9-7940X CPU.

Table 3. Results on EN-SAS dataset, all result values multiplied by 10^2. The best baseline results are underlined and the best overall results are in **bold**.

Method	*science*		*biology*		*English*		*Overall*	
	Q1,2,10 avg.		*Q5,6 avg.*		*Q3,4,7–9 avg.*		*10-que. avg.*	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
bert-pre-lstm	18.47	25.00	8.94	16.96	21.29	29.85	16.23	23.94
bert-pre-fine	20.89	26.47	9.18	14.62	23.46	29.01	17.84	23.36
bert-fine-fine	17.29	22.96	7.80	13.15	19.91	25.89	15.00	20.67
(rub text)-fine-fine	19.61	25.35	8.02	13.37	22.86	28.60	16.83	22.44
bert$_{wwm}$-pre-lstm	17.71	24.01	9.94	16.40	20.28	28.41	15.98	22.94
bert$_{wwm}$-pre-fine	19.27	24.69	10.48	14.38	22.24	28.08	17.33	22.38
bert$_{wwm}$-fine-fine	16.46	21.03	7.74	13.51	19.74	26.04	14.65	20.19
(rub text)$_{wwm}$-fine-fine	18.80	24.14	7.74	13.17	22.04	27.94	16.19	21.75
$sim^{(s)}_{sco}$-MeL-mlp	16.57	21.10	6.40	12.22	20.01	25.65	14.33	19.66
$sim^{(s)}_{sco}$-MeL$_{wwm}$-mlp	**15.48**	**19.83**	**5.73**	**11.37**	**18.82**	**24.89**	**13.34**	**18.70**
label-MeL-mlp	18.18	23.02	7.19	13.14	23.07	27.64	16.15	21.27
label-MeL$_{wwm}$-mlp	15.84	20.12	5.94	11.48	21.53	25.17	14.44	18.92

Table 4. Results on JP-SAS dataset, all result values multiplied by 10^2.

Method	*Q1*		*Q2*		*Q3*		Pre-training	Scoring
	MAE	RMSE	MAE	RMSE	MAE	RMSE	Time	Time
bert-pre-lstm	1.59	6.24	4.27	10.67	4.90	10.82	39 m	59 m
bert-pre-fine	4.51	7.83	9.40	13.54	7.59	12.52		6 m 10 s
bert-fine-fine	3.39	5.45	5.39	8.82	5.48	9.67	4 m 40 s	
(rub text)-fine-fine	3.45	6.16	6.22	10.62	6.57	11.24	5 min 20 s	
bert$_{wwm}$-pre-lstm	**1.41**†	5.54	3.71	9.55	4.78	10.44	37 m	5 h 22 m
bert$_{wwm}$-pre-fine	2.93	6.18	5.89	10.12	6.64	11.43		9 m 10 s
bert$_{wwm}$-fine-fine	2.09	5.55	5.13	9.18	5.24	9.86	7 m 50 s	
(rub text)$_{wwm}$-fine-fine	3.35	7.10	6.06	10.59	7.27	12.77	10 min 10 s	
$sim^{(s)}_{sco}$-MeL-mlp	2.02	5.03	3.97	8.39	4.74	**9.31***	3 m 40 s	30 s
$sim^{(s)}_{rub}$-MeL-mlp	1.61	4.91	3.81	**8.30****	**4.67***	9.36		
$sim^{(c)}_{rub}$-MeL-mlp	1.57	**4.83****	**3.62**†	8.36	4.77	9.43		
label-MeL-mlp	2.26	5.72	4.98	9.41	5.69	10.65	4 min 50 s	
half-data	2.25	5.46	5.00	9.60	5.70	10.47	2 m 10 s	
fourth-data	2.36	5.69	5.77	10.40	6.47	11.51	1 m	

Note: (1) We only show the results of MeL using *"bert-jp-base-wwm"* encoder in this table.
(2) We performed one-way ANOVA test to compare our results with the best baseline's.
*: sig $<$ 0.1, **: sig $<$ 0.05, †: $0.1 < \text{sig} < 0.2$

5 Scoring Results and Analysis

The scoring results on the two datasets are presented in Tables 3 and 4. It is clear that our method, *sim-MeL-mlp*, outperformed previous ASAS methods across most questions. The only exception is the MAE result of JP-SAS Q1. However, our approach significantly improved RMSE results on the JP-SAS dataset with 10x faster than *bert-fine-fine* in prediction time, and outpaced *bert-pre-lstm* by over 300x speed with comparable MAE accuracy. This demonstrates the substantial efficiency advantage of our method when processing large datasets.

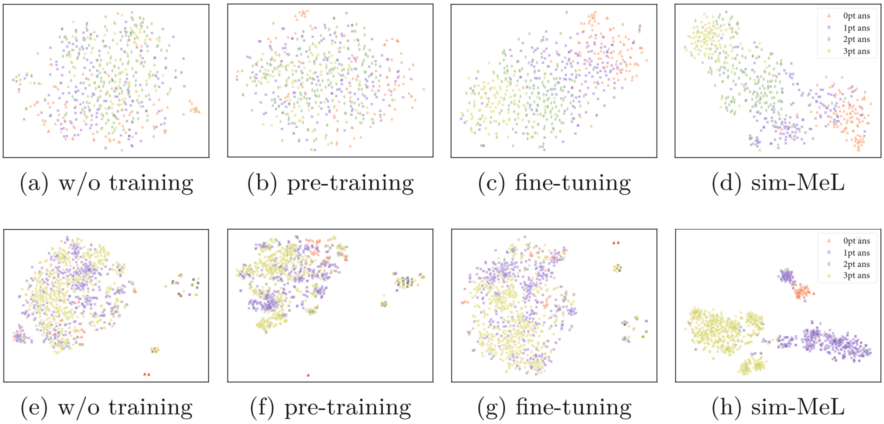
Furthermore, on the EN-SAS dataset, aside from the significant advantage of our method, we found that the two-stage fine-tuning method *bert-fine-fine*, which always conducted supervised learning, obtained better results than pre-training-based methods. We suspect that this could be due to the insufficient data and the large variance in contents among questions from the three subjects in EN-SAS, which led to a reduction in the effectiveness of pre-training. In contrast, on the JP-SAS dataset, the facts that the data for pre-training solely includes answers from Japanese reading subject, and with the availability of substantial data, all methods achieved better scoring accuracy. This time, the pre-training-based methods secured all the best baseline MAE results.

From another perspective, although *label-MeL* that used score labels directly as metrics also delivered commendable results, it was not as effective as our *sim-MeL*. This gap was especially noticeable on the JP-SAS dataset that expects higher accuracy due to its abundance of data, which highlights our claims about the importance of giving a more nuanced and comprehensive estimation of answer similarity as the metric. Additionally, the baseline method *(rub text)-fine-fine* that added rubric rules to answer texts did not achieve better results than methods without them, except in the case of biology in EN-SAS. We attribute this to the detailed rubric rules in biology (e.g., *bio-Q5: "List and describe four major steps involved in protein synthesis"; Rubric 3 points: "Four key elements..."*), compared to the vague requirements in other subjects such as English (e.g., *Eng-Q3, rubric 2 points: "The response demonstrates an exploration or development of the ideas..."*), which offer little help in further shaping answer features. The existence of those more precise and less variable answers in biology also led to all methods performing the best than in English.

Lastly, it can be concluded that using BERT with whole word masking (wwm) can improve scoring accuracy, helping *bert-pre-lstm* obtain superior MAE results on JP-SAS, and also gave significant improvement to *bert-fine-fine* on the less data-rich EN-SAS dataset. We consider the reason as the unique training ways of *bert-wwm*, which treats specific answer segments, such as proper nouns, as a whole, leading to a better comprehension of the phrases. Moreover, it appears that connecting the pre-trained BERT with an RNN structure might be a more effective approach, as it generally outperformed the *bert-pre-fine* model on both datasets. In summary, our method presents a promising balance between accuracy and efficiency, and holds significant implications for building a practical and efficient ASAS system.

6 Answer Embedding Space Analysis

To provide an intuitive understanding of the representation optimization effect of our approach, we present a 2D visualization of answer embedding spaces using t-SNE technique. Specifically, we compare the embedding spaces obtained from the encoder trained with our *sim-MeL* method to those obtained from the MLM



(a) w/o training (b) pre-training (c) fine-tuning (d) sim-MeL

(e) w/o training (f) pre-training (g) fine-tuning (h) sim-MeL

Fig. 4. Visualization of answer embedding spaces on EN-SAS-Q1 (upper) and JP-SAS-Q2 (bottom) test set

pre-training, and MSE fine-tuning method. The answers being embedded come from the test set of EN-SAS-Q1 and JP-SAS-Q2 (first 1,000 answers) and were all encoded with the "bert-wwm" encoder.

From the results shown in Fig. 4-(b) and (f), we can observe that traditional MLM pre-training gave only minor changes to the embedding distributions, providing less distinguished semantic information for downstream scoring. This is as expected since MLM is self-supervised. Meanwhile, as demonstrated in Figs. 4-(c) and (g), even after once supervised fine-tuning, the answer representations of different scores remained distributed with very little separation in the embedding spaces. This outcome observation coincides with our previous claim and prediction as depicted in Fig. 1-(b). Finally, with the use of the proposed *sim-MeL* approach, we noted that the answer embedding distributions transformed into a well-separated form, featuring much clearer inter-answer separation. We believe this provides a better condition for downstream scoring to achieve the best results even with a simple linear scoring network.

7 Conclusion

In this paper, we addressed the often-overlooked problem of answer representation optimization in ASAS studies, proposing a metric learning-based pre-training method to create distinct and separated answer embeddings. Using defined answer similarities based on scores and rubrics, we achieved superior results over previous ASAS approaches in both scoring accuracy and efficiency.

The future work will mainly focus on refining the similarity definition methods. For instance, we may further distinguish between each individual scoring point to build a more accurate rubric-based similarity metric. We would also like to experiment with more complex metric learning structures, such as supervised contrastive learning [9], to further optimize the representation space, striving for more compact answer clusters to improve scoring results.

Acknowledgment. This work was supported in part by JST SPRING No. JPMJSP2136 and JSPS KAKENHI No. JP21H00907 and JP23H03511.

References

1. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic text scoring using neural networks. In: ACL, pp. 715–725 (2016)
2. Chen, H., He, B.: Automated essay scoring by maximizing human-machine agreement. In: EMNLP 2013, pp. 1741–1752 (2013)
3. Condor, A., Litster, M., Pardos, Z.: Automatic short answer grading with SBERT on out-of-sample questions. In: EDM, pp. 345–352 (2021)
4. Condor, A., Pardos, Z., Linn, M.: Representing scoring rubrics as graphs for automatic short answer grading. In: AIED, pp. 354–365 (2022)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
6. Dong, F., Zhang, Y.: Automatic features for essay scoring-an empirical study. In: EMNLP 2016, pp. 1072–1077 (2016)
7. Dong, F., Zhang, Y., Yang, J.: Attention-based recurrent convolutional neural network for automatic essay scoring. In: CoNLL, pp. 153–162 (2017)
8. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, pp. 84–92 (2015)
9. Khosla, P., et al.: Supervised contrastive learning. arXiv:2004.11362 (2020)
10. Larkey, L.S.: Automatic essay grading using text categorization techniques. In: ACM SIGIR, pp. 90–95 (1998)
11. Li, X., Yang, H., Hu, S., Geng, J., Lin, K., Li, Y.: Enhanced hybrid neural network for automated essay scoring. Expert. Syst. **39**(10), e13068 (2022)
12. Lun, J., Zhu, J., Tang, Y., Yang, M.: Multiple data augmentation strategies for improving performance on automatic short answer scoring. In: AAAI, pp. 13389–13396 (2020)
13. Luo, D., Su, J., Yu, S.: A BERT-based approach with relation-aware attention for knowledge base question answering. In: IJCNN, pp. 1–8. IEEE (2020)
14. Mayfield, E., Black, A.W.: Should you fine-tune BERT for automated essay scoring? In: Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 151–162 (2020)
15. Oord, A.V.D., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
16. Phandi, P., Chai, K.M.A., Ng, H.T.: Flexible domain adaptation for automated essay scoring using correlated linear regression. In: EMNLP, pp. 431–439 (2015)
17. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv:1908.10084 (2019)
18. Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., Arora, R.: Pre-training BERT on domain resources for short answer grading. In: EMNLP 2019, pp. 6071–6075 (2019)
19. Viji, D., Revathy, S.: A hybrid approach of weighted fine-tuned BERT extraction with deep Siamese Bi-LSTM model for semantic text similarity identification. Multimedia Tools Appl. **81**(5), 6131–6157 (2022)
20. Wang, T., Funayama, H., Ouchi, H., Inui, K.: Data augmentation by rubrics for short answer grading. J. Nat. Lang. Process. **28**, 183–205 (2021)

21. Wang, Z., Lan, A.S., Waters, A.E., Grimaldi, P., Baraniuk, R.G.: A meta-learning augmented bidirectional transformer model for automatic short answer grading. In: EDM, pp. 667–670 (2019)
22. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**(2), 207–244 (2009)
23. Yang, R., Cao, J., Wen, Z., Wu, Y., He, X.: Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In: Findings of EMNLP 2020, pp. 1560–1569 (2020)
24. Zhu, X., Wu, H., Zhang, L.: Automatic short-answer grading via BERT-based deep neural networks. IEEE Trans. Learn. Technol. **15**(3), 364–375 (2022)

Prompting Generative Language Model with Guiding Augmentation for Aspect Sentiment Triplet Extraction

Kun Huang[1,2], Yongxiu Xu[1,2](✉), Xinghua Zhang[1,2], Wenyuan Zhang[1,2], and Hongbo Xu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{huangkun,xuyongxiu,zhangxinghua,zhangwenyuan,hbxu}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Aspect Sentiment Triplet Extraction (ASTE) is a challenging task in natural language processing concerning the automatic extraction of (*aspect term*, *opinion term*, *sentiment polarity*) triplets from a given text. Current end-to-end generative methods achieved high results by treating it as a sequence generation task with a generative pretrained language mode (e.g., T5). However, these architectures usually suffered from the objective gaps between the pre-training tasks and fine-tuning tasks, leading to suboptimal results. Further more, they can only provide information on what is a valid triplet, but no explicit guidance on what is not a triplet, which can not fully capture the correlation between aspects and opinions. To address above issues, we propose the *generative prompt* to bridge the gap between pre-training and fine-tuning of generative pre-trained language model via text infilling task. And we propose *guiding augmentation*, which drops the aspect or opinion in the sentence by depicting a tree structure to generate diverse similar sentences and new target sequences. In this way, the main differences between these augmented samples are the dropped aspect or opinion term, and the model can understand the ASTE task knowledge better through the explicit variant constraints. Experimental results confirm that our method outperforms previous state-of-the-art (SOTA) methods on four public ASTE datasets.

Keywords: ASTE · Prompt · Data Augmentation

1 Introduction

Aspect-based Sentiment Triplet Extraction(ASTE) is a branch of the Aspect-based Sentiment Analysis(ABSA), which aims at identifying and analyzing aspects, their corresponding opinion terms and sentiment in text. As shown in Fig. 1, given the sentence "Service was slow, but the people were friendly", the output would be two triplets: (service, slow, negative) and (people, friendly, positive).

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 249–261, 2024.
https://doi.org/10.1007/978-981-99-7022-3_22

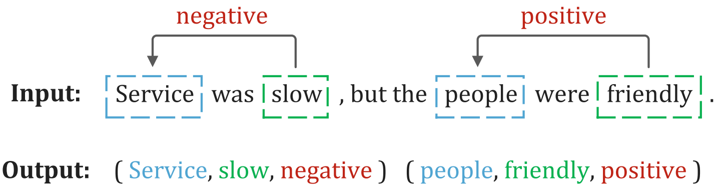

negative
positive
Input: Service was slow , but the people were friendly .
Output: (Service, slow, negative) (people, friendly, positive)

Fig. 1. An example for ASTE task

Early ABSA research employed a pipeline approach [7] that decomposed ASTE into multiple subtasks. However, this approach suffered from cascading errors. Thus, recent research has focused on end-to-end methods to address the issue. Zhang et al. [17] introduced a generative model, GAS, which employed manually constructed templates to transform target triplets into a target sequence. The model was subsequently trained to generate this target sequence. Zhang et al. [16] further proposed a paraphrasing paradigm that integrated sentiment elements in natural language, making better use of the semantic information of sentiment labels. Mao et al. [6] used beam search to independently decode each triplet, alleviating the issue of false dependencies between triplets during the generation process in generative models.

Although existing generative methods have achieved promising results, they still suffered from two main issues: 1) **The fine-tuning process of generative models for ASTE task differs from the pre-training task:** current generative ASTE methods typically use generative models (such as T5 [12] and BART [4]) as their backbone model, which pretrains the model with sequence-to-sequence mask language modeling (MLM) task, while fine-tuning the model with a natural language generation task. Therefore, there may be a gap between the pre-training and fine-tuning processes in the ASTE task. 2) **Can not fully capture the correlation between aspects and opinions:** generative models need to transform triplets into target sentence. Although the transformed sentences can make use of the semantic information of the labels, some task information may be lost during the transformation process. This can make the model only know what is an aspect-opinion pair but ignore what is not an aspect-opinion pair. As a result, generative models may generate non-existent triplets during the generation process.

To address the aforementioned issues, we propose two techniques: 1) **generative prompt**: Petroni et al. [8] used BERT to complete downstream tasks in a cloze-style manner and achieved promising results. Inspired by this, we propose the generative prompt. As shown in Fig. 2 (c), we concatenate the generative prompt to the end of the input sentence. The special tokens "<extra_id_0>", "<extra_id_1>", and "<extra_id_2>" in the prompt represent the aspect term, opinion term, and sentiment polarity, respectively. We use text infilling to allow the model to generate the aspect term, opinion term and sentiment, then obtaining the triplets. 2) **guiding augmentation**: To enable the model to learn task knowledge, we propose the guiding augmentation. We construct a tree structure as shown in Fig. 3, where each aspect and opinion information are dropped at different levels. We treat the leaf nodes in the tree as independent samples and input them into the model. Based on the remaining task-related information in the leaf nodes, we construct new target sequences and train the model to generate them. By constraining the model with the remaining information and the new target sequences, we guide the model to learn task knowledge, including each sentiment element and its relationship in the ASTE task.

Our contributions are as follows:

1. We propose the **generative prompt**, which bridges the gap between pretraining process of generative language model and fine-tuning on ASTE task.
2. We develop the **guiding augmentation** to drop different sentiment element information and train the model to generate new target sequences based on the remaining information, thereby guiding model to comprehensively understand the sentiment triplet knowledge.
3. We conduct experiments on public ASTE datasets. The average F1 score is improved by 2.54%, 0.99%, and 1.14% compared to SOTAs (COM-MRC [15], BDTF [18], Span-Bidirectional [2]).

2 Our Approach

2.1 Task Definition

Given a sentence $S = \{w_1, w_2, ..., w_n\}$, where w_i represents the i^{th} word in the sentence, the objective of ASTE is to extract all sentiment triplets T. Each triplet T is defined as (a, o, s), where a represents the aspect terms in the sentence that contain sentiment, o represents the opinion terms associated with the aspect terms, and s represents the sentiment polarity expressed in the aspect terms, $s \in \{positive, negative, neutral\}$.

2.2 Generative Prompt

Current generative methods [17] typically used manually designed templates to transform sentiment triplets into target sentence, and then fine-tune generative model to generate the target sentence to obtain triplets. As shown in Fig. 2(b), given sentence "Service was slow, but the people were friendly", let model generate target sentence "(Service, slow, negative); (people, friendly, positive)". However, as shown in Fig. 2(a), in the pre-training of T5[12], given the sentence "Thank you for inviting me to your party last week", replace the words "for inviting" and "last" with special tokens, and ask the model to generate them. Thereby, there is a gap between the fine-tuning and the pre-training task used in language models (e.g., T5 [12]).

Inspired by the Paraphrase [16], which used the natural language to combine sentiment elements in a triplet to enhance the model's understanding of the

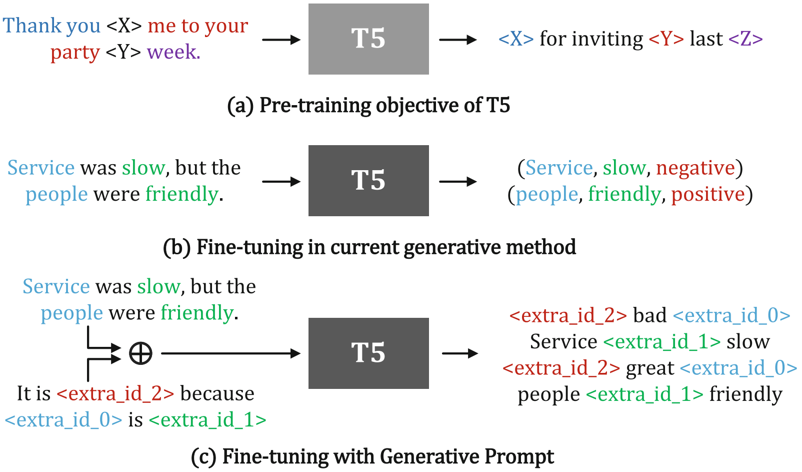

Thank you <X> me to your party <Y> week.
T5
<X> for inviting <Y> last <Z>
(a) Pre-training objective of T5
Service was slow, but the people were friendly.
T5
(Service, slow, negative)
(people, friendly, positive)
(b) Fine-tuning in current generative method
Service was slow, but the people were friendly.
⊕
It is <extra_id_2> because <extra_id_0> is <extra_id_1>
T5
<extra_id_2> bad <extra_id_0> Service <extra_id_1> slow <extra_id_2> great <extra_id_0> people <extra_id_1> friendly
(c) Fine-tuning with Generative Prompt

Fig. 2. The pre-training and fine-tuning of T5

semantic relationship between each element. We have developed a generative prompt method, as shown in Fig. 2(c). The prompt consists of the phrase:

$$\text{It is <extra\_id\_2> because <extra\_id\_0> is <extra\_id\_1>} \tag{1}$$

where "<extra_id_0>", "<extra_id_1>", and "<extra_id_2>" representing aspect terms, opinion terms, and sentiment polarity, respectively. Then train the model to generate aspect terms, opinion terms and sentiment polarity through text infilling.

As for target sequence, we follow the Paraphrase [16], and define a mapping function for each sentiment element in a sentiment triplet (a, o, s): 1) $Pa(a) = x_{at}$, where x_{at} represents the aspect terms in the sentence; 2) $Po(o) = x_{ot}$, where x_{ot} represents the opinion terms in the sentence; and 3) $Ps(s) = x_{sp}$, where x_{sp} is a word in the semantic space to which sentiment s is mapped, and $x_{sp} \in \{great, bad, ok\}$, indicating the sentiment polarity. Afterwards, we insert these values into the following template paradigm:

$$\text{<extra\_id\_2>}\, x_{sp}\ \text{<extra\_id\_0>}\, x_{at}\ \text{<extra\_id\_1>}\, x_{ot} \tag{2}$$

It is worth noting that when a sentence contains multiple triplets, we concatenate them based on the order of appearance of the aspect terms and opinion terms in the sentence to obtain the target sequence y.

2.3 Guiding Augmentation

Some ASTE-related information in the triplets may be lost after the conversion process. Given the sentence: "Service was slow, but the people were friendly", the previous generative methods could only inform the model that there are

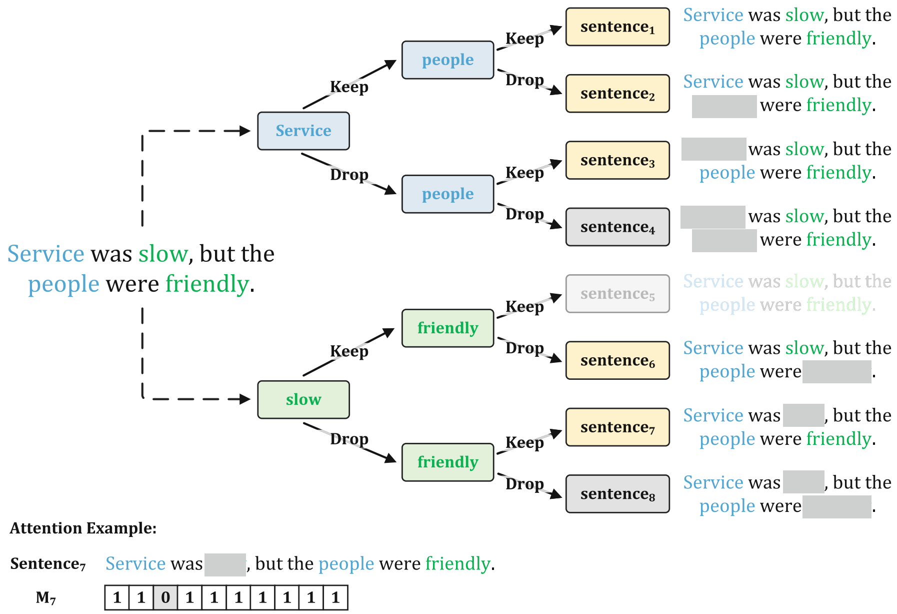

Service was slow, but the people were friendly.
Service
slow
Keep
Drop
people
people
friendly
friendly
sentence₁
Service was slow, but the people were friendly.
sentence₂
Service was slow, but the were friendly.
sentence₃
was slow, but the people were friendly.
sentence₄
was slow, but the were friendly.
sentence₅
Service was slow, but the people were friendly.
sentence₆
Service was slow, but the people were .
sentence₇
Service was , but the people were friendly.
sentence₈
Service was , but the people were .
Attention Example:
Sentence₇ Service was , but the people were friendly.
M₇ 1 1 0 1 1 1 1 1 1 1

Fig. 3. TreeMask: using the binary tree structure, we drop the aspect information from the original sentence, where each non-leaf node represents an aspect. Each edge from a non-leaf node to its child is labeled to indicate whether the information represented by the non-leaf node should be dropped or retained. Each leaf node (except for the gray nodes) represents a sample. The same process applies to the opinion.

two aspect-opinion pair: “(service, slow)” and “(people, friendly)”, but couldn’t inform the model that “(service, friendly)” is non-existent aspect-opinion pair. As a result, the model may generate completely non-existent triplets (either the aspect term or the opinion term in the triplet does not overlap with the aspect and opinion terms in the aspect-opinion pairs of the ground-truth triplets). To address this issue, we propose guiding augmentation. This approach enables the model to utilize different task-related information within the similar samples to generate different target sequences. By doing so, the model can understand what to generate and what not to generate in the absence of certain aspect or opinion information, thereby guiding the model to understand ASTE task.

As shown in Fig. 3, We propose a TreeMask operation for guiding augmentation with a binary tree structure. Each non-leaf node in the tree corresponds to an aspect or opinion in S, and the leaf nodes represent the sub-spans of S that remain after the aspect or opinion information has been dropped. Each edge from a non-leaf node to its child is assigned a label of “keep” or “drop”, based on whether the child contains the aspect or opinion information.

By using the labels assigned to the edges, we generate an attention mask M that indicates which token w_i in S should be dropped out. Specifically, for each non-leaf node’s edge labeled “drop”, we set the attention weights of all token

w_i in the corresponding sub-spans to zero in M. For each non-leaf node's edge labeled "keep", we set the attention weights of all tokens in the corresponding sub-spans to one in M. And the attention weights of other tokens are one.

$$M_k = \begin{cases} 0, & \text{if } w_i \text{ is dropped} \\ 1, & \text{others} \end{cases} \tag{3}$$

Here, k represents the k^{th} leaf node. As shown in Fig. 3, the attention weight of "slow" is zero in $sentence_7$, others are one. Considering a sentence should contain at least one complete triplet in real data, we discard the gray nodes that all aspects or opinions have been dropped. And we ignore the $sentence_5$ because it is repeated with $sentence_1$. Then we use the original sequence S and the k^{th} attention mask M_k to generate the target sequence y_k.

$$y_k = LM(S, M_k) \tag{4}$$

LM is generative language model. We construct target sequence y_k for each leaf node according to the paradigm rules described in Sect. 2.2, based on the complete triplet information contained in that node. As show in Fig. 3, in $sentence_7$, there is a complete triplet "(people, friendly, POS)", so the target sequence y_7 is "<extra_id_2>great<extra_id_0> people<extra_id_1> friendly".

2.4 Training and Inference

Training. As show in Fig. 4, We use the T5 model as the backbone, for each input sample j in dataset, we obtain a input set X_j with guiding augmentation and a corresponding target sequence set Y_j. We then train the LM using X_j and Y_j as input and output pairs, respectively. The model is trained using a negative log-likelihood loss function L_θ, defined as:

$$L_\theta = -\sum_{j=1}^{N} \sum_{(y_k,x_k)\in(Y_j,X_j)} \sum_{i=1}^{n} \log P_\theta(y_{k,i}|y_{k,<i}, p, x_k) \tag{5}$$

$$x_k = (S, M_k) \tag{6}$$

where θ is the model parameters, N is the number of training examples, p is the prompt in Sect. 2.2, and the n is the length of the y_k, S is the original sentence in the $sample_j$, M_k is the k^{th} attention mask obtained by TreeMask.

Inference. In the inference phase, we do not use the guiding augmentation. We just concatenate the prompt p with the input sentence and feed it directly into the trained model to generate a sequence. We then decode the generated sequence according to the paradigm rules to obtain the triplets.

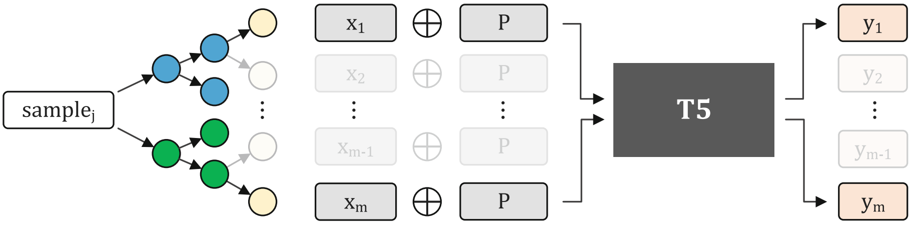


Fig. 4. Model architecture, P is the prompt in Sect. 2.2, m represents the total number of sentences obtained from sample j after the TreeMask operation.

3 Experiments

3.1 Datasets

We conduct experiments on the public datasets, sourced from SemEval14 [11], SemEval15 [9] and SemEval16 [10]. Our data was downloaded from GAS [17].

3.2 Baseline Methods

We classify the baseline methods into four categories: 1) **Pipeline methods:** Peng et al. [7] divided the triplet extraction into two stages, and BMRC [2] and COM-MRC [15] divided the task into multiple rounds reading comprehension. 2) **Table-filling methods:** GTS [13] designed an inference method that constructed a relationship table and inferred triplets through table labeling. BDTF [18] used region detection in computer vision to identify the potential relationship regions and then determine the type of relationship. 3) **Span-based methods** enumerated all spans and identified the type of each span, while determining the relationship between them to infer the triplet. Span-ASTE [14] introduced a dual-channel span pruning strategy to reduce the high computational cost caused by span enumeration. Span-Bidirectional [2] used KL divergence to encourage similar spans to be as far apart as possible in feature space and employed a dual-channel decoding approach to improve model performance. 4) **Generative methods:** GAS[17] transformed triplets into target sequences, treating ASTE as a sequence generation task. Paraphrase [16] mapped sentiment elements to words in the semantic space, fully utilizing the semantic information of the labels.

3.3 Implementation Details

We use T5-base [12] as the backbone model, with a maximum of 30 epochs during training. We verify the model's performance on the validation set, and test the best model on the test set. The experimental hyperparameters are set according to Paraphrase [16], with a learning rate of 1e-4 for rest14 and laptop14, and 3e-4 for rest15 and rest16. We use the AdamW optimizer [5] with a batch size of 16, and train the model using an RTX Titan. We conduct experiments on each dataset using five different random seeds, and then average the results.

3.4 Main Result

Table 1. The main experimental result

Model	Rest14	Lap14	Rest15	Rest16	average F1
	F1	F1	F1	F1	
Pipeline method					
two-stage [7]	51.46	42.87	52.32	54.21	50.21
BMRC [2]	67.99	57.82	60.02	65.75	62.90
COM-MRC [15]	72.01	60.17	64.53	71.57	67.07
Table-filling method					
GTS-BERT [13]	67.50	54.36	60.15	67.93	62.49
BDTF [18]	74.35	61.74	66.12	72.27	68.62
Span-based method					
Span-ASTE [14]	71.85	59.38	63.27	70.26	66.19
Span-Bidirectional [2]	74.34	62.65	64.82	72.08	68.47
Generative method					
GAS [17]	72.16	60.78	62.10	70.10	66.29
Paraphrase [16]	72.03	61.13	62.56	71.70	66.86
Ours	**74.38**	**64.86**	**65.66**	**73.55**	**69.61**

As shown in Table 1, our approach has demonstrated its effectiveness by outperforming the current state-of-the-art (SOTA) methods: COM-MRC [15], BDTF [18], and Span-Bidirectional [2], on all four datasets. Specifically, our average F1 score is 2.54%, 0.99%, and 1.14% higher than the three SOTA methods, respectively, further supporting the effectiveness of our approach in addressing the ASTE task. Notably, compared to Paraphrase [16], which is like our method, our F1 scores increased by 2.35%, 3.73%, 3.1%, and 1.85% on the four datasets, demonstrating the effectiveness of our generative prompt and guiding augmentation in improving the performance of generative models.

3.5 Ablation Study

Our ablation experiments on the four datasets are presented in Table 2. The "w/o aug" refers to experiments without guiding augmentation, and "w/o p" refers to experiments without the prompt (in Sect. 2.2). The results show a significant drop in performance without guiding augmentation or generative prompt, indicating their effectiveness. Moreover, the results suggest that guiding augmentation is more effective than the generative prompt.

Furthermore, the inclusion of guiding augmentation result in a higher precision increase than recall on all four datasets, with a precision increase of 4.18%,

Table 2. Ablation Study: "*w/o* aug" means that the experimental method does not include guiding augmentation, while "*w/o* p" means that the experimental method does not include the prompt(in Sect. 2.2).

	Rest14			Lap14			Rest15			Rest16		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Ours	75.34	73.44	**74.38**	69.32	60.92	**64.86**	65.48	65.86	**65.66**	73.12	75.02	**73.55**
w/o aug	71.16	71.23	71.19	64.78	62.33	63.52	60.34	64.58	62.38	69.65	73.39	71.46
w/o p	74.00	72.31	73.14	69.35	60.63	64.49	64.61	64.33	64.47	72.88	73.54	73.20

4.54%, 5.13%, and 3.47%, respectively. This indicates that the model generated triplets more accurately with guiding augmentation, and that guiding augmentation can suppress the generation of non-existent triplets. Overall, our ablation experiments provide evidence for the effectiveness of guiding augmentation and generative prompt in improving the performance of our approach.

3.6 More Analysis

Table 3. The experimental results are recorded separately for single-aspect and multi-aspect cases. "Single" refers to the case where there is only one aspect mentioned in the sentence, while "Multi" refers to the case where there are multiple aspects mentioned. "*w/o* aug" represents the case without guiding augmentation method.

	Rest14			Lap14			Rest15			Rest16		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Single	68.20	79.32	**73.34**	66.57	73.26	**69.75**	61.35	70.00	**65.39**	64.98	75.31	**69.76**
w/o aug	63.40	76.70	69.42	60.64	71.52	65.63	56.40	69.90	62.43	61.40	74.48	67.30
Multi	77.68	71.90	**74.68**	71.37	54.62	61.86	69.49	62.69	**65.88**	79.20	73.23	**76.08**
w/o aug	73.76	69.79	71.72	67.73	57.59	62.23	64.32	60.51	62.34	75.88	72.73	74.27

Our guiding augmentation tends to construct shorter target sequences with more single aspect cases. To demonstrate that the performance improvement is not due to generating shorter sequences, we conduct an analysis of single and multiple aspect cases on the four datasets, as presented in Table 3. The results show that adding guiding augmentation improves the F1 and recall for all single aspect cases, as well as most multiple aspect cases on all datasets. Moreover, the precision score for both single and multiple aspect cases improved after the addition of guiding augmentation, further supporting the effectiveness of our approach. These results indicate that our approach is not simply generating shorter sequences but rather capturing the knowledge relevant to the ASTE task.

To verify that our method can suppress the generation of non-existent triplets (either the aspect term or the opinion term in the triplet does not overlap with

Table 4. The proportion of non-existent triplets among the incorrect triplets: "*w/o* aug" refers to the case without guiding augmentation. **The lower value, the better.**

	Rest14	Lap14	Rest15	Rest16
Ours	52.54	46.09	46.67	49.31
w/o aug	57.65	54.46	55.07	54.46

the aspect and opinion terms in the aspect-opinion pairs of the ground-truth triplets) by the model, we count the proportion of non-existent triplets among the incorrect triplets generated by the model with/without Guiding Augmentation. The results are shown in Table 4. We find that the proportion of completely non-existent triplets decreased by 5.12%, 8.37%, 8.40%, and 8.58% after adding Guiding Augmentation. This experiment confirms that Guiding Augmentation can suppress the generation of non-existent triplets by the model.

3.7 Case Study

Table 5. Case study: ✔ means the tirplet is right, ✘ means the triplet is wrong.

Exp 1: the atmosphere is very nice, and a welcome escape from the rest of the SI mall.		
Ground Truth	Ours (*w/o* aug)	Ours
(atmosphere, nice, POS)	(atmosphere, nice, POS)✔ (atmosphere, welcome, POS)✘	(atmosphere, nice, POS)✔
Exp 2: The service is always great, and the owner walks around to make sure you enjoy.		
Ground Truth	Ours (*w/o* aug)	Ours
(service, great, POS)	(service, great, POS)✔ (owner, enjoy, POS)✘	(service, great, POS)✔
Exp 3: The service was dreadfully slow and a smile would have been nice...		
Ground Truth	Ours (*w/o* aug)	Ours
(service, dreadfully slow, NEG)	(service, slow, NEG)✘	(service, slow, NEG)✘

In Table 5, we compare the results with and without guiding augmentation. In the first example, given the sentence "the atmosphere is very nice, and a welcome escape from the rest of the SI mall", due to the closeness between "atmosphere" and "welcome" and the presence of the word "and" between them, the model without guiding augmentation wrongly interprets "welcome" as the opinion of "atmosphere", and generates an extra triplet ("atmosphere", "welcome", "POS"), where "welcome" is not an opinion term that modifies "atmosphere". With guiding augmentation, the model is able to discern it is not opinion. In the second example, given the sentence "The service is always great, and the owner walks around to make sure you enjoy", without guiding augmentation, the model interprets "owner" and "enjoy" in the second half of the sentence as a pair, and generates an extra triplet (“owner”, “enjoy”, “POS”). With the guiding augmentation, the model is able to correctly identify that they are not a pair. This demonstrates that guiding augmentation can help the model identify what is not an aspect and opinion term. However, in the third example, while there are adverbs (e.g., “dreadfully”) in the aspect terms or opinion terms, the model is unable to correctly extract the aspect-opinion pair due to incorrect boundaries of extracted aspect or opinion terms. This is a challenging problem in the ASTE task, and this is also our future research work.

4 Related Work

Recently, the ASTE task has received widespread attention. Peng et al. [7] proposed the concept of ASTE and decomposed it into two steps: 1)extract aspect terms and opinion terms respectively; 2) determine whether there is a sentimental relationship between the two. Another mainstream pipeline method for the ASTE task is to use the MRC [2] mechanism to extract the sentiment elements in the triplets sequentially. However, the pipeline method suffers from cascading errors, so Jing et al. [3], Chen et al. [1], Mao et al. [6] and others proposed the end-to-end methods. The end-to-end method is divided into two categories: discriminative and generative. Discriminative methods generally model the relationship between tokens and then infer triplets through table annotations [13], or enumerate each span in the sentence [14], determine its type and the relationship between spans, then infer triplets. Generative methods [16] generally transform triplets based on rule templates into the target sequence, and then let the model generate the target sequence to infer triplets.

5 Conclusion

This paper proposes two techniques, Generative Prompt and Guiding Augmentation, to complete the ASTE task using a generative model. Generative Prompt is designed to bridge the gap between downstream fine-tuning and pre-training in generative pretrained models. Guiding Augmentation drops information about aspects or opinions in the sentence to generate multiple similar samples, which enables the model to acquire knowledge related to the ASTE task. Experiments are conducted on four public datasets, and the results show that our proposed method outperforms the current SOTA methods and the approximate method Paraphrase. The ablation study demonstrates that both Generative Prompt and Guiding Augmentation are effective. Moreover, we analyze the errors produced by the model and find that our proposed method can prevent the model from generating completely non-existent triplets. However, we also find that the model still faces challenges in terms of boundary errors, which is our future work.

Acknowledgement. This work was supported by Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02040400).

References

1. Chen, H., Zhai, Z., Feng, F., Li, R., Wang, X.: Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 2974–2985 (2022)
2. Chen, S., Wang, Y., Liu, J., Wang, Y.: Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12666–12674 (2021)
3. Jing, H., Li, Z., Zhao, H., et al.: Seeking common but distinguishing difference, a joint aspect-based sentiment analysis model. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3910–3922 (2021)
4. Lewis, M., Liu, Y., 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, pp. 7871–7880 (2020)
5. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
6. Mao, Y., Shen, Y., Yang, J., Zhu, X., Cai, L.: Seq2Path: generating sentiment tuples as paths of a tree. In: Findings of the Association for Computational Linguistics: ACL 2022, pp. 2215–2225 (2022)
7. Peng, H., Xu, L., Bing, L., Huang, F., Lu, W., Si, L.: Knowing what, how and why: a near complete solution for aspect-based sentiment analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8600–8607 (2020)
8. Petroni, F., Rocktäschel, T., Riedel, S., et al.: Language models as knowledge bases? 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. 2463–2473 (2019)
9. Pontiki, M., Galanis, D., et al.: SemEval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic evaluation (SemEval 2015), pp. 486–495 (2015)
10. Pontiki, M., et al.: SemEval-2016 task 5: aspect based sentiment analysis. In: ProWorkshop on Semantic Evaluation (SemEval-2016), pp. 19–30. Association for Computational Linguistics (2016)
11. Pontiki, M., Papageorgiou, H., et al.: SemEval-2014 task 4: aspect based sentiment analysis. In: SemEval 2014, p. 27 (2014)
12. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 5485–5551 (2020)
13. Wu, Z., Ying, C., Zhao, F., Fan, Z., Dai, X., Xia, R.: Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2576–2585 (2020)
14. Xu, L., Chia, Y.K., Bing, L.: Learning span-level interactions for aspect sentiment triplet extraction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 1: Long Papers), pp. 4755–4766 (2021)
15. Zhai, Z., Chen, H., Feng, F., Li, R., Wang, X.: COM-MRC: a context-masked machine reading comprehension framework for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 3230–3241 (2022)

16. Zhang, W., Deng, Y., Li, X., Yuan, Y., Bing, L., Lam, W.: Aspect sentiment quad prediction as paraphrase generation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9209–9219 (2021)
17. Zhang, W., Li, X., Deng, Y., Bing, L., Lam, W.: Towards generative aspect-based sentiment analysis. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (vol. 2: Short Papers), pp. 504–510 (2021)
18. Zhang, Y., et al.: Boundary-driven table-filling for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 6485–6498 (2022)

Prompting GPT-3.5 for Text-to-SQL with De-semanticization and Skeleton Retrieval

Chunxi Guo, Zhiliang Tian$^{(\boxtimes)}$, Jintao Tang$^{(\boxtimes)}$, Pancheng Wang, Zhihua Wen,
Kang Yang, and Ting Wang$^{(\boxtimes)}$

College of Computer, National University of Defense Technology, Changsha, China
{chunxi,tianzhiliang,tangjintao,wangpancheng13,zhwen,yangkang,
tingwang}@nudt.edu.cn

Abstract. Text-to-SQL is a task that converts a natural language question into a structured query language (SQL) to retrieve information from a database. Large language models (LLMs) work well in natural language generation tasks, but they are not specifically pre-trained to understand the syntax and semantics of SQL commands. In this paper, we propose an LLM-based framework for Text-to-SQL which retrieves helpful demonstration examples to prompt LLMs. However, questions with different database schemes can vary widely, even if the intentions behind them are similar and the corresponding SQL queries exhibit similarities. Consequently, it becomes crucial to identify the appropriate SQL demonstrations that align with our requirements. We design a de-semanticization mechanism that extracts question skeletons, allowing us to retrieve similar examples based on their structural similarity. We also model the relationships between question tokens and database schema items (i.e., tables and columns) to filter out scheme-related information. Our framework adapts the range of the database schema in prompts to balance length and valuable information. A fallback mechanism allows for a more detailed schema to be provided if the generated SQL query fails. Ours outperforms state-of-the-art models and demonstrates strong generalization ability on three cross-domain Text-to-SQL benchmarks.

Keywords: Large language model · Text-to-SQL · Prompt learning

1 Introduction

Text-to-SQL tasks aim to transform natural language questions (NLQ) into structured query language (SQL), enabling users without expertise in database querying to retrieve information from a database [1,2]. Considering that databases are used in various scenarios involving different domains (e.g., education, financial systems), researchers have adapted encoder-decoder architecture [3,4], which eliminates the need for domain-specific knowledge through end-to-end training. To train the model, these approaches require diverse and extensive training data, which can be prohibitively expensive [5].

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 262–274, 2024.
https://doi.org/10.1007/978-981-99-7022-3_23

Large pre-trained language models (LLMs) (e.g., GPT-3 [6] and Codex [7]) encompass more extensive data and parameters than traditional pre-trained language models (e.g., BERT [8], RoBERTa [9], BART [10] and T5 [11]) and exhibit superior performance on a variety of tasks, including Text-to-SQL. Rajkumar et al. [12] and Liu et al. [13] evaluate LLMs' performance in Text-to-SQL in zero- and few-shot settings. Cheng et al. [14] present a neural-symbolic framework that maps the input to a program, which incorporates symbolic components into LLMs. However, many studies found that LLMs perform worse than traditional non-LLM-based approaches in Text-to-SQL [3,15,16]. As the existing LLMs are not designed for understanding the syntax and semantics of SQL commands, it is challenging for them to accurately generate complex SQL commands (e.g. *SELECT*, *WHERE*, *AVG*, *DESC*). To accurately map out these SQL commands, it is essential to distinguish the question intention. Intention in Text-to-SQL tasks refers to the collection of query-related specifications and directives that encompass the desired scope, criteria, and actions to be performed on a database. This concept encompasses various desired result set attributes, including data volume, sorting sequence, and filtering prerequisites. For example, the skeleton corresponds to the question *"What are the names of the singers who are not French?"* is *"What are the [MASK] of the [MASK] who are not [MASK]?"*, whose intention is to query a term with a conditional constraint.

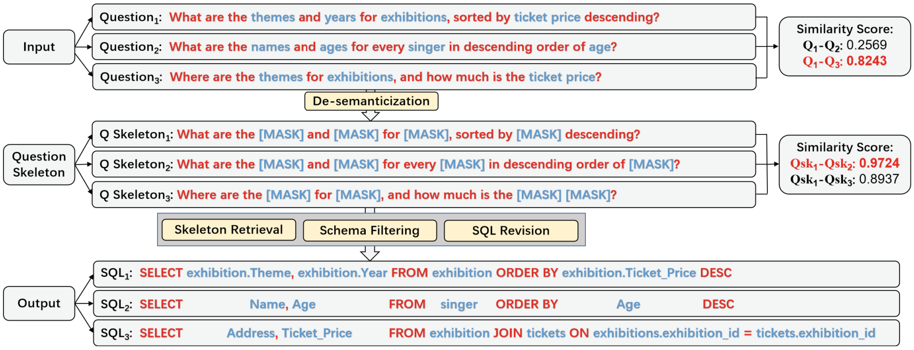

Input
Question$_1$: What are the themes and years for exhibitions, sorted by ticket price descending?
Question$_2$: What are the names and ages for every singer in descending order of age?
Question$_3$: Where are the themes for exhibitions, and how much is the ticket price?
Similarity Score:
Q$_1$-Q$_2$: 0.2569
Q$_1$-Q$_3$: 0.8243
De-semanticization
Question Skeleton
Q Skeleton$_1$: What are the [MASK] and [MASK] for [MASK], sorted by [MASK] descending?
Q Skeleton$_2$: What are the [MASK] and [MASK] for every [MASK] in descending order of [MASK]?
Q Skeleton$_3$: Where are the [MASK] for [MASK], and how much is the [MASK] [MASK]?
Similarity Score:
Qsk$_1$-Qsk$_2$: 0.9724
Qsk$_1$-Qsk$_3$: 0.8937
Skeleton Retrieval
Schema Filtering
SQL Revision
Output
SQL$_1$: SELECT exhibition.Theme, exhibition.Year FROM exhibition ORDER BY exhibition.Ticket_Price DESC
SQL$_2$: SELECT Name, Age FROM singer ORDER BY Age DESC
SQL$_3$: SELECT Address, Ticket_Price FROM exhibition JOIN tickets ON exhibitions.exhibition_id = tickets.exhibition_id

Fig. 1. Comparison of three examples (i.e. question, question skeleton, SQL). The first example is similar to the second in terms of question intention (sorting), and to the third in terms of vocabulary of the questions. Note that the intention of the third question is to get two attribute items and there may be a table join. We aim to obtain SQL queries with the same commands (i.e. *ORDER BY*, *DESC*) in the prompt. Compared with the full question similarity score, the question skeleton increases the absolute value as well as the relative ranking.

LLMs are proven to fast adapt to new paradigms with few-shot examples [17–19]. We argue that LLMs probably quickly learn to follow some demonstration examples of SQL generation, even if the SQL generation involves multiple SQL commands and nested clauses.

In this paper, we propose an LLM-based Text-to-SQL framework that retrieves a few demonstration examples to prompt the LLM according to the skeleton of the input question. Notice that questions with different database schemes may be distinct since questions contain much scheme-related information (i.e. the blue texts in Fig. 1's upper side), even if they have similar intention and SQL queries. It can be difficult for the model to retrieve helpful examples. To solve this issue, we design a de-semanticization mechanism to extract skeletons of questions. We retrieve similar SQL demonstrations, which share similar question skeletons with the input question. Besides, during de-semanticization, we model the relationships between question tokens and scheme items to filter out the scheme items related to the question tokens. In this way, we achieve schema linking concerning the question-scheme relevance. Finally, we adaptively control the range of the database schema in prompts to balance length and valuable information. Through a fallback mechanism, the model receives a more detailed schema if the generated SQL query needs revision. Our framework excels compared to commercial Text-to-SQL engines like AI2sql[1] by emphasizing transparent, flexible algorithms for better performance and deeper insights into the conversion process, whereas comparisons to large models like GPT-4 might not be entirely fair due to data volume discrepancies.

Our contributions are as follows: (1) We propose an LLM-based framework for Text-to-SQL tasks that retrieves similar examples to augment prompts for LLMs. (2) We design a de-semanticization mechanism that effectively removes scheme-related information for retrieving texts with similar skeletons. (3) Our method surpasses the SOTA models and exhibits strong generalization.

2 Related Work

The evolution of SQL generation techniques showcases a progression from encoder-decoder architectures [2] to LLM-based solutions.

Encoder-Based SQL Generation. Guo et al. [20] introduced IRNET, utilizing attention-based Bi-LSTM to encode and an intermediate representation-based decoder for SQL prediction. Afterwards, graph-based encoders were integrated to enhance input representations [21,22]. Works such as RATSQL [1], LGESQL [23], R2SQL [24], SDSQL [3], S2SQL [25], and STAR [26] focused on refining structural reasoning by explicitly modeling relationships between schemas and questions. Notably, GRAPHIX-T5 [4] overcame prior limitations by incorporating graph representation learning into the encoder. Simultaneously, RASAT [16] augmented T5 with structural insights by introducing edge embeddings into multi-head self-attention.

Decoder-Based SQL Generation. We categorize the methods into four distinct groups: Sequence-based methods like BRIDGE [27] and PICARD [15] directly translate natural language queries into SQL queries token by token. Template-based methods, represented by X-SQL [28] and HydraNet [29], utilize predefined templates to guide SQL generation, ensuring structural coherence. Stage-based methods, exemplified by GAZP [30] and RYANSQL [31],

[1] https://app.ai2sql.io.

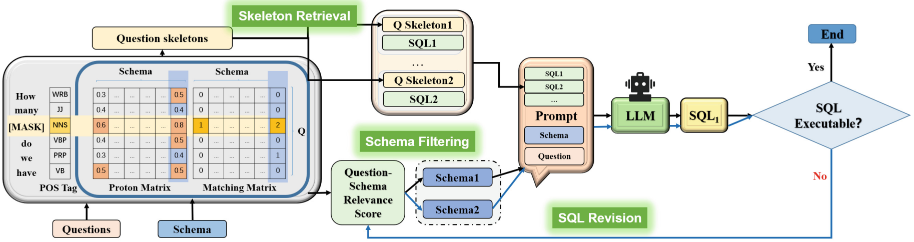


Fig. 2. The overview of our framework. The grey box on the left shows the details of the de-semantization process. The rest is the process of prompt construction. We revise the SQL following the blue line. (Color figure online)

involve establishing a coarse-grained SQL framework, subsequently employing slot-filling methodologies to complete missing details within the framework. Lastly, hierarchical-based methods, such as IRNet [20] and RAT-SQL [1], adopt a hierarchical approach to tackle NLQ-to-SQL translation.

LLM-Based SQL Generation. Recently, LLM-based models are now prominent options for this task. Unlike full-data fine-tuned models, LLM-based models can achieve good performance with just a few unsupervised in-context exemplar annotations [5]. Yu et al. [32] introduced a method to classify and cluster SQL queries based on question characteristics. Inspired by some retrieval-related research [33–35], we retrieve SQL examples with the same intention as a demonstration, thereby enhancing the comprehension of the diverse operators and their respective applications. Our approach improves the LLM's performance to generate valid and accurate SQL queries.

3 Methodology

Our framework consists of two modules as shown in Fig. 2: (1) **Question De-semanticization** (Sect. 3.1) removes tokens that are semantically related to the domain and preserves the question skeletons, which represent the question's intentions. (2) **LLM-Based Adjustable Prompting** (Sect. 3.2) involves using the SQL demonstrations with the same intention and corresponding database schema to create prompts that guide the LLM in generating SQL queries.

Given a natural language question Q and the database schema $S = \langle T, C \rangle$, the goal of Text-to-SQL tasks is to generate the corresponding SQL P. Here the question $Q = \left(q_1, q_2, \ldots, q_{|Q|}\right)$ is a sequence of words. The database schema consists of tables $T = \left(t_1, t_2, \cdots, t_{|T|}\right)$, and columns $C = \left(c_1, c_2, \cdots, c_{|C|}\right)$.

3.1 Question De-semanticization

We remove question tokens that are semantically related to the database schema (i.e., table, column) and obtain the question skeletons, which represent the question's intentions. In this way, from a matching perspective, by eliminating the schema-related information from the question, we can better match examples with similar question intent when retrieving.

There are two steps involved in this process: the first step is schema linking, which finds the question tokens related to the schema. Meanwhile, we obtain the relevance of the question tokens to the schema. As a second step, we mask the tokens and obtain the skeletons of the questions. In the first step, we cannot determine how relevant the tokens are to the schema, so we design schema-related detection (Sect. 3.1.1), token matching (Sect. 3.1.2) and part-of-speech tagging (Sect. 3.1.3) strategies to find relevant tokens.

3.1.1 Schema-Related Detection

We use a masking technique to identify correlations between question words and their corresponding database schema. Concretely, we calculate the similarity between question tokens and schema items. There are three steps:

(1) Masking and Concatenation. We concatenate a question Q and schema (i.e., table T, column C) into one long sequence. We also mask each token of the question in the sequence to generate a series of sequences. Formally, we obtain a series of sequences as follows, where [CLS] and [SEP] denote classification and sentence separation, respectively:

$$[CLS]q_1q_2\cdots q_{|Q|}[SEP][CLS]t_1\cdots t_{|T|}[SEP][CLS]c_1\cdots ,c_{|C|}[CLS]$$

$$[CLS][MASK]q_2\cdots q_{|Q|}[SEP][CLS]t_1\cdots ,t_{|T|}[SEP][CLS]c_1\cdots ,c_{|C|}[CLS]$$

$$\cdots$$

Later, we put all sequences into a pre-trained language model to obtain deep contextualized representations. We denote h_j^s as the representation of schema item s_j and $h_{j\backslash q_i}^s$ as the representation if the question token q_i is masked out.

(2) Representation Transformation. We utilize the standard Poincaré ball [36]-a unique model of hyperbolic spaces to project the representations. We get the hyperbolic representations, denoted as $\tilde{h}$, using the following method:

$$\tilde{\mathrm{h}} = g_0(\mathrm{~h}) = \tanh(\|\mathrm{h}\|)\frac{\mathrm{h}}{\|\mathrm{h}\|}. \tag{1}$$

The hyperbolic space provides a suitable geometry for modeling the hierarchy [36]. The hyperbolic space has a property called a negative curve which allows for a more efficient representation of the hierarchy, enabling our method to capture long-term dependencies and the overall sentence structure [37]. Furthermore, the hyperbolic space has a greater power to represent than the Euclidean space low-dimensional space, thus allowing for a more efficient representation of sentences with semantic hierarchy.

(3) Correlation Measurement. We measure the correlation between the question token q_i and the schema item s_j in the hyperbolic space. Concretely, we elicit the correlation between question tokens and schema items from a pre-trained language model based on the Poincaré distance matrix [37].

By computing d_p on each pair of tokens (q_i, s_j), we get the Proton matrix $\mathrm{D_p} \in \mathbb{R}^{|Q|\times|S|}$ as follows:

$$\mathrm{D_p}_{i,j} = 2\tanh^{-1}\left(\left\|-\tilde{h}^s_{j\backslash q_i} \oplus \tilde{h}^s_j\right\|\right), \tag{2}$$

where $\oplus$ is the Möbius addition [36], $\tilde{h}^s_j$ represents the embedding of the schema item s_j, and $\tilde{h}^s_{j\backslash q_i}$ represents the embedding if the question token q_i is masked out. Both $\tilde{h}^s_j$ and $\tilde{h}^s_{j\backslash q_i}$ are hyperbolic representation defined in Eq. (1).

We argue that the sequence concatenating all tables and columns represents the domain knowledge of the example, which consists of the vocabulary of the domain/scenario. We mask each token in the question sequence to detect whether it is relevant to the domain. This measure works by masking the token and checking if it causes a significant shift in the vector representation of the entire sequence. If the shift is beyond a pre-defined threshold, we consider the masked token to be important for the sequence. This means that the token is strongly correlated with the meaning expressed by the sequence.

3.1.2 Token Matching

To discover the question tokens closely related to the scheme, we match each question token and each schema item (i.e., table names, column names, and values) with two kinds of explicit information in the input: name-based and value-based matching.

Name-based matching identifies direct lexical matches between each question token and each schema item. If a sub-sequence of the question sequence $q_{i\ldots j}$ matches the schema names s_j, score one point for matching similarity. While value-based matching detects possible value correspondences within the query. If the question word q_i is equal to specific values v_j in the database, where v_j represents the set of values in the j^{th} column of the corresponding table or column, score one point for the corresponding matching similarity. We define matrix $M_m \in \mathbb{R}^{|Q|\times|S|}$ to represent the question-schema matching similarity:

$$M_{m_{i,j}} = \begin{cases} 2, & q_{i\ldots j} \subseteq s_j \wedge q_i = v_j \\ 1, & q_{i\ldots j} \subseteq s_j \vee q_i = v_j \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

So far, coupled with the previously calculated Proton matrix $\mathrm{D_p}$, we get the question-schema relevance score, which measures the probability that the schema items will be used to compose the SQL query. We define the relevance score matrix $R \in \mathbb{R}^{|Q|\times|S|}$ as follows:

$$R = \mathrm{D_p} + \beta \cdot M_m, \tag{4}$$

where β determines the relative influence of these two strategies.

3.1.3 Part-of-Speech Tagging

We perform part-of-speech (POS) tagging on questions to improve the recognition of the question skeleton.

Formally, we tag the question Q with POS analysis to obtain a set of POS tags $t_1, t_2, \ldots, t_n$, where t_i is the POS tag of a token q_i. Then we generate the lexical matrix $P \in \mathbb{R}^{|Q|}$ for each token q_i based on its POS tag t_i as follows:

(1) If t_i is a noun or a number, then P_i is assigned the value α.
(2) Otherwise, P_i is assigned the value 0.

POS information is crucial for constructing a question skeleton as it aids in comprehending the sentence's structure and the grammatical roles of the words.

Incorporating the three strategies mentioned above, we obtain a question relevance score ${Q_{\text{sco}}}_i$ for each question token q_i using the following equation:

$$ {Q_{\text{sco}}}_i = \frac{1}{2}\left(\frac{1}{n}\sum_{j=1}^{|S|} R_{ij} + P_i\right), \quad \forall i \in 1, \ldots, |Q|. \tag{5}$$

We generate the question skeleton based on Q_{sco} and τ, where τ is a hyperparameter that controls the minimum relevance score required for a token.[2]

After calculating the R (in Sect. 3.1.2) and P, we remove domain-relevant tokens of the questions and generate the de-semanticized question skeletons, which allows for better retrieval of examples where the question intentions are more consistent and more applicable to the in-context demonstration.

3.2 LLM-Based Adjustable Prompting

To construct prompts for the LLM to generate new SQL queries, we utilize the SQL queries obtained from the question skeleton and the relevant database schema, which are filtered by the question-schema relevance score. Then we revise the SQL queries via a fallback mechanism, which adjusts the schema range. The prompt we designed consists of three parts as shown in Fig. 2.

3.2.1 *k*NN-Based Skeleton Retrieval

We retrieve k-NN examples based on the new question skeleton. We project de-semanticized question skeletons into a vector space and retrieve k-NN examples corresponding to the new question skeleton. We use cosine similarity to measure the text vectors. The new question skeleton serves as the key for the retrieval process, and the returned value consists of the k-NN examples. Questions with the same intention can aid in generating SQL queries by sharing common structures and requiring comparable SQL queries to extract information from a database. Recognizing patterns and similarities between questions enables the language model to produce suitable SQL queries for a given inquiry.

[2] If q_{sco} is below the threshold of τ, we retain the original question token; otherwise, we replace it with the pre-defined [MASK] token.

3.2.2 Schema-Relevance Filtering

We utilize the filtered database schema items to generate SQL queries. We get the schema with a scaled-down range by the relevance scores between the question and the schema. We apply a threshold θ to filter out less relevant schema items based on the question-schema relevance score obtained in Sect. 3.1. Specifically, we only consider schema items with scores higher than the θ. with a scaled-down schema range, we prompt the LLM to generate the SQL queries. Narrowing the schema range prevents irrelevant schema from interfering with the model's response to the current question.

3.2.3 Fallback Revision

We propose a fallback mechanism to revise and regenerate SQL queries, in cases where the LLM outputs a message indicating SQL generation failure or when the generated SQL query cannot be executed successfully. We first check if the generated SQL query is valid and can be executed on the database. If the query fails, we retrieve the complete database schema and use it to revise the SQL query. This ensures that the revised SQL query is valid and can be executed on the database. We then pass the revised SQL query to the LLM for further processing. To avoid generating an infinite loop of fallbacks, we set a maximum number of fallback attempts. If the maximum number of fallback attempts is reached and the SQL query still cannot be generated, we terminate the process.

4 Experiment

4.1 Experimental Setup

Datasets. In our study, we perform experiments on three widely recognized benchmark datasets: (1) **Spider** [38] covers a diverse range of 138 domain databases, offering a large-scale evaluation platform. (2) **Spider-Syn** [39] is a modified version of Spider that introduces difficulty by replacing explicit question-schema alignments with synonymous phrasing. (3) **Spider-DK** [40] is an augmentation of Spider, incorporating artificial domain knowledge to further test model adaptability and comprehension.

Evaluation. Valid SQL (VA) measures the percentage of SQL queries that are executed without any errors. **Execution accuracy (EX)** measures the accuracy of the execution results by comparing them with the standard SQL query. **Test-suite accuracy (TS)** [41] measures the effectiveness of the distilled test suite in achieving high code coverage for the database through execution, which can serve as a better proxy for semantic accuracy. Note that we do not rely on the mainstream exact match accuracy metric (EM), as SQL queries that serve the same purpose may be expressed in different ways. EM is tailored to a limited style of the dataset and serves as an intermediate solution evaluation metric for Text-to-SQL tasks.

Baselines. Full-data fine-tuned models: **PICARD** [15] employs incremental parsing to constrict auto-regressive decoders in language models; **RASAT** [16] enhances transformer models with relation-aware self-attention, combined with constrained auto-regressive decoders; and **RESDSQL** [3] introduces a novel framework featuring ranking-enhanced encoding and skeleton-aware decoding. LLM-based models: We select the **ChatGPT** [13] and **Codex** [12] baseline models for our study, as they are currently the top performers in evaluating Text-to-SQL capability using LLMs. Furthermore, our approach utilizes the latest **GPT-3.5** model, text-davinci-003. We all use the best performing with prompt engineering methodologies adapted to our respective models.

Experimental Setting. We use FAISS [42] to store and retrieve question skeletons. For hyperparameter settings, we assign $k = 8$, $\alpha = 0.9$, $\beta = 0.5$, $\tau = 0.6$, and $\theta = 0.4$. Our approach expands the database content beyond the limitations of the Text-to-SQL prompt used in the OpenAI demo website[3], which only contains table and column names. We re-formatted the prompt to achieve better results, though it differs from the format used in the official data training.

4.2 Main Results

Table 1. Comparison of the performance of our model and others on three datasets ("-" indicates the results are not available. The Codex model could not be reproduced due to an invalid Codex'api. The TS metric did not apply to the Spider-DK dataset).

Models\Datasets		SPIDER			SPIDER-SYN			SPIDER-DK		
		VA	EX	TS	VA	EX	TS	VA	EX	TS
Full-data Fine-tuned Models	PICARD	98.4	79.3	69.4	98.2	69.8	61.8	97.8	62.5	-
	RASAT	98.8	80.5	70.3	98.3	70.7	62.4	98.5	63.9	-
	RESDSQL-3B	**99.1**	84.1	73.5	98.8	76.9	66.8	98.8	66.0	-
LLM-based Models	GPT-3.5	87.0	57.2	56.7	83.1	39.3	39.2	88.6	48.8	-
	Codex [12]	91.6	67.0	55.1	-	-	-	-	-	-
	ChatGPT [13]	97.7	70.1	60.1	96.2	58.6	48.5	96.4	62.6	-
	Ours	99.0	**87.8**	**84.8**	**99.0**	**79.4**	**75.9**	**99.4**	**74.2**	-

We present a comparison between LLM-based models and full-data fine-tuned models which are SOTA as shown in Table 1. Ours outperforms all models in almost all evaluation metrics, except for the Spider dataset where the VA is only 0.1 worse than the next best model (RESDSQL-3B). LLM-based models may face difficulty generating SQL queries that conform to strict syntactical and semantic rules, resulting in lower VA scores, compared with the fine-tuned models. We further investigate the effectiveness of ours and compare it to the

[3] https://platform.openai.com/examples/default-sqltranslate.

models using LLMs directly. We observe that the improper utilization of LLMs is generally less effective in generating complex SQL commands. Additionally, the regular LLM-based models are confused with selecting the appropriate schema items required. On the contrary, ours demonstrates its effectiveness in generating accurate and semantically meaningful SQL queries.

4.3 Ablation Study

We investigate the contributions of various components of ours as shown in Table 2: (1) DESEM+P, which uses Poincaré distance for schema-related detection in de-semanticization; (2) DESEM-P+E, which uses Euclidean distance for schema-related detection in de-semanticization; (3) -DESEM, which retrieves questions without de-semanticization using cosine similarity; (4) -Skeleton Retrieval, which demonstrates random examples without retrieval; (5) -Schema Filtering, which uses the full range of schema without filtering; (6) -SQL Revision, which directly outputs the generated SQL without revision.

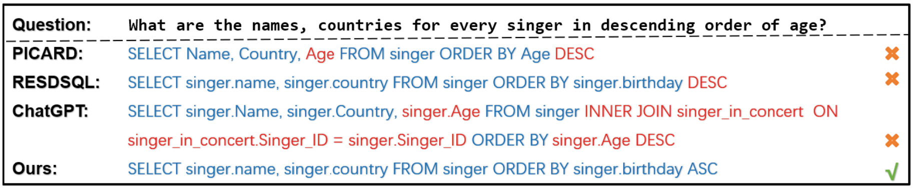
Table 2. Ablation study of different modules. "-" means not using that strategy, while "+" means using that strategy.

Methods/Datasets	SPIDER			SPIDER-SYN			SPIDER-DK		
	VA	EX	TS	VA	EX	TS	VA	EX	TS
DESEM +P (Ours)	**99.0**	**87.8**	**84.8**	**99.0**	**79.4**	**75.9**	**99.4**	**74.2**	**69.7**
DESEM -P+E	96.6	84.6	79.6	97.1	77.2	74.3	99.1	71.3	68.2
-DESEM	93.5	71.6	69.8	94.7	59.2	57.7	86.5	56.8	55.0
-Skeleton Retrieval	94.3	66.9	65.9	92.9	56.8	55.3	93.3	59.4	57.9
-Schema Filtering	96.7	79.6	78.3	97.3	73.6	71.5	98.5	68.6	64.3
-SQL Revision	98.5	81.2	78.2	95.0	69.9	67.3	97.6	71.8	67.1

Using Poincaré distance instead of Euclidean distance for schema detection (DESEM + P) improves efficacy. Removing de-semanticization (-DESEM) reduces accuracy, highlighting the importance of relevant examples for utilizing SQL components and expressing intent. Removing skeleton retrieval and comparing question sequences directly lowers performance, indicating domain knowledge's impact on the SPIDER-DK. SQL revision enhances generated SQL accuracy, as seen in higher VA scores of methods with revision. Schema filtering boosts performance by narrowing the schema range, even without revision.

4.4 Case Study

To illustrate our method, we show a comparison of predicted SQLs in Fig. 3 using PICARD, RESDSQL, ChatGPT, and our approach.



Question:	What are the names, countries for every singer in descending order of age?	
PICARD:	SELECT Name, Country, Age FROM singer ORDER BY Age DESC	✗
RESDSQL:	SELECT singer.name, singer.country FROM singer ORDER BY singer.birthday DESC	✗
ChatGPT:	SELECT singer.Name, singer.Country, singer.Age FROM singer INNER JOIN singer_in_concert ON singer_in_concert.Singer_ID = singer.Singer_ID ORDER BY singer.Age DESC	✗
Ours:	SELECT singer.name, singer.country FROM singer ORDER BY singer.birthday ASC	✓

Fig. 3. An illustrative case from Spider-DK [40]. Blue is the correct generation, and red is the wrong generation. (Color figure online)

It shows that PICARD and ChatGPT generate an extra column "Age", and the three models fall short of accurately capturing the desired sorting order. This is because the database schema only has a "birthday" column, and sorting by age is equivalent to sorting by "birthday" in ascending order. ChatGPT attempts to generate a more comprehensive SQL query by including an unnecessary JOIN operation with the singer_in_concert table. Ours considers the requirement of ordering the results, albeit in the opposite direction specified in the question. However, fine-tuned models like PICARD and RESDSQL may struggle with complex questions due to limitations in learned patterns and structures.

5 Conclusion and Future Work

We propose a Text-to-SQL generation framework that prompts LLMs with few retrieved demonstrations. A limitation of ours is over-reliance on LLMs's ability for SQL code generation. LLMs with certain capabilities can work well with our method. Future research will focus on external knowledge reasoning, as well as efficiency when dealing with large databases. Our approach can be generalized to knowledge base question answering and code generation tasks.

References

1. Wang, B., Shin, R., Liu, X., Polozov, O., Richardson, M.: RAT-SQL: relation-aware schema encoding and linking for text-to-SQL parsers. ACL (2020)
2. Cai, R., Xu, B., Zhang, Z., Yang, X., Li, Z., Liang, Z.: An encoder-decoder framework translating natural language to database queries. In: IJCAI (2018)
3. Li, H., Zhang, J., Li, C., Chen, H.: Decoupling the skeleton parsing and schema linking for text-to-SQL. arXiv:2302.05965 (2023)
4. Li, J., Hui, B., et al.: Graphix-T5: mixing pre-trained transformers with graph-aware layers for text-to-SQL parsing. arXiv:2301.07507 (2023)
5. Zhao, W.X., Zhou, K., Li, J., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
6. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. In: NIPS, vol. 33, pp. 1877–1901 (2020)
7. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.D.O., et al.: Evaluating large language models trained on code. arXiv:2107.03374 (2021)

8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2018)
9. Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized bert pre-training approach with post-training. In: CCL, pp. 1218–1227 (2021)
10. Lewis, M., Liu, Y., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL (2020)
11. Raffel, C., Shazeer, N., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR **21**, 5485–5551 (2020)
12. Rajkumar, N., Li, R., Bahdanau, D.: Evaluating the text-to-SQL capabilities of large language models. arXiv:2204.00498 (2022)
13. Liu, A., Hu, X., Wen, L., Yu, P.S.: A comprehensive evaluation of ChatGPT's zero-shot text-to-SQL capability. arXiv:2303.13547 (2023)
14. Cheng, Z., Xie, T., Shi, P., et al.: Binding language models in symbolic languages. In: ICLR (2023)
15. Scholak, T., Schucher, N., Bahdanau, D.: Picard: parsing incrementally for constrained auto-regressive decoding from language models. In: EMNLP (2021)
16. Qi, J., Tang, J., He, Z., et al.: RASAT: integrating relational structures into pretrained Seq2Seq model for text-to-SQL. In: EMNLP, pp. 3215–3229 (2022)
17. Lee, Y.J., Lim, C.G., Choi, H.J.: Does GPT-3 generate empathetic dialogues? A novel in-context example selection method and automatic evaluation metric for empathetic dialogue generation. In: COLING, pp. 669–683 (2022)
18. Su, H., Kasai, J., et al.: Selective annotation makes language models better few-shot learners. arXiv:2209.01975 (2022)
19. Rubin, O., Herzig, J., Berant, J.: Learning to retrieve prompts for in-context learning. In: NAACL, pp. 2655–2671 (2022)
20. Guo, J., et al.: Towards complex text-to-SQL in cross-domain database with intermediate representation. In: ACL, pp. 4524–4535 (2019)
21. Bogin, B., Berant, J., Gardner, M.: Representing schema structure with graph neural networks for text-to-SQL parsing. In: ACL (2019)
22. Chen, Z., et al.: ShadowGNN: graph projection neural network for text-to-SQL parser. In: NAACL (2021)
23. Cao, R., Chen, L., et al.: LGESQL: line graph enhanced text-to-SQL model with mixed local and non-local relations. In: ACL (2021)
24. Hui, B., Geng, R., Ren, Q., et al.: Dynamic hybrid relation exploration network for cross-domain context-dependent semantic parsing. In: AAAI (2021)
25. Hui, B., Geng, R., Wang, L., et al.: S2SQL: injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers. In: ACL, pp. 1254–1262 (2022)
26. Cai, Z., Li, X., Hui, B., Yang, M., Li, B., et al.: Star: SQL guided pre-training for context-dependent text-to-SQL parsing. In: EMNLP (2022)
27. Lin, X.V., Socher, R., Xiong, C.: Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In: EMNLP, pp. 4870–4888 (2020)
28. He, P., Mao, Y., Chakrabarti, K., Chen, W.: X-SQL: reinforce schema representation with context. arXiv:1908.08113 (2019)
29. Lyu, Q., Chakrabarti, K., Hathi, S., Kundu, S., Zhang, J., Chen, Z.: Hybrid ranking network for text-to-SQL. arXiv preprint arXiv:2008.04759 (2020)
30. Zhong, V., Lewis, M., Wang, S.I., Zettlemoyer, L.: Grounded adaptation for zero-shot executable semantic parsing. In: EMNLP, pp. 6869–6882 (2020)
31. Choi, D., Shin, M.C., Kim, E., Shin, D.R.: Ryansql: recursively applying sketch-based slot fillings for complex text-to-SQL in cross-domain databases. CL **47**(2), 309–332 (2021)

32. Yu, W., Guo, X., Chen, F., Chang, T., Wang, M., Wang, X.: Similar questions correspond to similar SQL queries: a case-based reasoning approach for text-to-SQL translation. In: Sánchez-Ruiz, A.A., Floyd, M.W. (eds.) ICCBR 2021. LNCS (LNAI), vol. 12877, pp. 294–308. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86957-1_20
33. Tian, Z., Bi, W., Li, X., Zhang, N.L.: Learning to abstract for memory-augmented conversational response generation. In: ACL, pp. 3816–3825 (2019)
34. Song, Y., et al.: Retrieval bias aware ensemble model for conditional sentence generation. In: ICASSP, pp. 6602–6606. IEEE (2022)
35. Wen, Z., et al.: Grace: gradient-guided controllable retrieval for augmenting attribute-based text generation. In: Findings of ACL 2023, pp. 8377–8398 (2023)
36. Ganea, O., Bécigneul, G., Hofmann, T.: Hyperbolic neural networks. In: NIPS (2018)
37. Chen, B., et al.: Probing bert in hyperbolic spaces. arXiv:2104.03869 (2021)
38. Yu, T., Zhang, R., et al.: Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In: EMNLP (2019)
39. Gan, Y., Chen, X., Huang, Q., Purver, M., et al.: Towards robustness of text-to-SQL models against synonym substitution. In: ACL (2021)
40. Gan, Y., Chen, X., Purver, M.: Exploring underexplored limitations of cross-domain text-to-SQL generalization. In: EMNLP (2021)
41. Zhong, R., Yu, T., Klein, D.: Semantic evaluation for text-to-SQL with distilled test suites. In: EMNLP, pp. 396–411 (2020)
42. Johnson, J., Douze, M., Jegou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**, 535–547 (2019)

QURG: Question Rewriting Guided Context-Dependent Text-to-SQL Semantic Parsing

Linzheng Chai[1], Dongling Xiao[2], Zhao Yan[2], Jian Yang[1](✉), Liqun Yang[1], Qian-Wen Zhang[2], Yunbo Cao[2], and Zhoujun Li[1]

[1] State Key Lab of Software Development Environment, Beihang University, Beijing, China
{challenging,jiaya,lqyang,lizj}@buaa.edu.cn
[2] Tencent Cloud Xiaowei, Beijing, China
{zhaoyan,yunbocao}@tencent.com

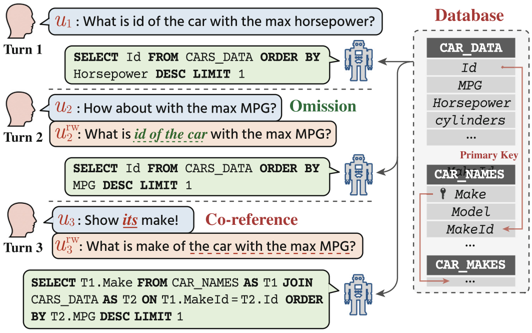
Abstract. Context-dependent Text-to-SQL aims to translate multi-turn natural language questions into SQL queries. Despite various methods have exploited context-dependence information implicitly for contextual SQL parsing, there are few attempts to explicitly address the dependencies between current question and question context. This paper presents QURG, a novel **QU**estion **R**ewriting **G**uided approach to help the models achieve adequate contextual understanding. Specifically, we first train a question rewriting model to complete the current question based on question context, and convert them into a rewriting edit matrix. We further design a two-stream matrix encoder to jointly model the rewriting relations between question and context, and the schema linking relations between natural language and structured schema. Experimental results show that QURG significantly improves the performances on two large-scale context-dependent datasets SParC and CoSQL, especially for hard and long-turn questions.

Keywords: Semantic Parsing · Context-dependent Text-to-SQL · Question Rewriting

1 Introduction

The past decade has witnessed increasing attention on text-to-SQL semantic parsing task, which aims to map natural language questions to SQL queries. Previously, works have mainly concentrated on the context-independent text-to-SQL task [32], which translates single questions to SQL queries. The key to solving context-independent text-to-SQL is to model the relationships between questions and schema. Recent works have made great progress [2,14,22,24,25] by employing pre-train language model. Compared with the context-independent text-to-SQL task, the context-dependent text-to-SQL task faces more challenges,

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 275–286, 2024.
https://doi.org/10.1007/978-981-99-7022-3_24


u_1 : What is id of the car with the max horsepower?

```
SELECT Id FROM CARS_DATA ORDER BY
Horsepower DESC LIMIT 1
```

u_2 : How about with the max MPG?

u_2^{rw}: What is *id of the car* with the max MPG?

```
SELECT Id FROM CARS_DATA ORDER BY
MPG DESC LIMIT 1
```

u_3 : Show *its* make!

u_3^{rw}: What is make of the car with the max MPG?

```
SELECT T1.Make FROM CAR_NAMES AS T1 JOIN
CARS_DATA AS T2 ON T1.MakeId = T2.Id ORDER
BY T2.MPG DESC LIMIT 1
```

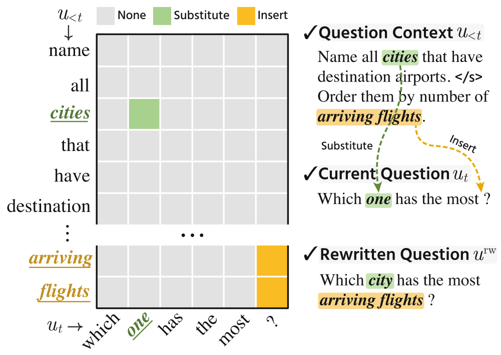
Fig. 1. An example of the context-dependent Text-to-SQL task with the phenomenon of co-reference and omission. u_t^{rw} denotes the rewritten question of the current question u_t at t-th conversation turn.

that not only need to consider the relationship between natural language questions and the schema, but also the relationship between the current question and question context.

In a multi-turn scenario, as shown in Fig. 1, current questions may contain two contextual phenomena: *co-reference* and *omission* which are heavily associated with the historical context, meanwhile the question context may also contain information irrelevant to current questions. Thus models are required to selectively leverage contextual information to correctly address the user's intention of the current questions.

Previous works [7,19,31,35] on context-dependent text-to-SQL typically model the context dependencies in a simple way that feeds the concatenation of the current question, question context and schema into a neural networks encoder. Several works directly leverage historical generated SQL [34,35] or track interaction states associated with historical SQL [1,23] to enhance the current SQL parsing. However, these works neglect the explicit guidance on resolving contextual dependency. [3] is the first attempt at context-dependent text-to-SQL task by question rewriting, but this approach relies on in-domain QR annotations and complex algorithms to obtain the rewritten question data.

To address the above limitations, we propose QURG, a novel QUestion Rewriting Guided approach, which consists of three steps: 1) rewriting the current question into self-contained question and further converting it into a rewriting edit matrix; 2) jointly representing the rewriting matrix, multi-turn questions, and schema; 3) decoding the SQL queries. Firstly, we train and evaluate the QR model on the out-of-domain dataset CANARD [6] and initialize the QR model with a pre-trained sequence generator for more precise rewritten questions. Secondly, inspired by [16], we propose to integrate rewritten results into the text-to-SQL task in the form of a rewriting relation matrix between question and context. We observed that directly replacing or concatenating original input with rewritten question may mislead the model for correctly SQL parsing. The reason is the unavoidable noise in rewritten questions and some questions



Name all cities that have destination airports. </s> Order them by number of arriving flights.

Which one has the most ?

Which city has the most arriving flights ?

Fig. 2. An example of rewriting edit matrix. Given rewritten question u_t^{rw}, we convert it to relations between current question u_t and question context $u_{<t}$.

are semantically complete and do not need to be rewritten. Taking Fig. 2 as an example, the rewriting edit matrix denotes the relations between question and context words, these relations could clearly guide the model in solving long-range dependencies.

Furthermore, we propose a two-stream relation matrix encoder based on the relation-aware Transformer (RAT) [20] to jointly model the rewriting relation features between the current question and the context, and the schema linking relation features between multi-turn question and database schema. Finally, we aggregate the representations from the two relation matrix encoders to generate current SQL queries.We evaluate our proposed QURG on two large-scale cross-domain context-dependent benchmarks: SParC [33] and CoSQL [30]. We summarize the contributions of this work as follows:

- We present a novel context-dependent text-to-SQL framework QURG that explicitly guides models to resolve contextual dependencies.
- Our framework incorporates rewritten questions in a novel way that explicitly represents multi-turn questions through rewriting relation matrix and two-stream relation matrix encoder.
- Experimental results show that QURG achieves comparable performance to recent state-of-the-art works on two context-dependent text-to-SQL datasets.

2 Related Work

Natural language processing tasks [9–11] have grown significantly with the development of pre-train language model. The task aims to map natural language questions to database-related SQL queries. Spider [32] is a widely evaluated cross-domain context-independent dataset and numerous works [2,14,18,20,22,26,27,29] have shown that modeling the relation between question and schema can effectively improve performance on Spider.

In the face of the *co-reference* and *omission* in multi-turn questions, context-dependent text-to-SQL task is more challenging. Several works [23,34,35] utilize previously generated SQL queries to resolve long-range dependency and improve the parsing accuracy. Some works [1,7] use graph neural networks to jointly encode multi-turn questions and schema. Others [12,31] propose auxiliary state switch prediction tasks to model multi-turn question relations. [19] simply constrain the auto-regressive decoders of super large pre-trained language models T5-3B. In this work, we propose to adopt the rewriting matrix to explicitly model the relationships between the current question and context.

3 Preliminaries

In this section, we first formalize the context-dependent Text-to-SQL task, and then we introduce the relation-aware Transformer (RAT) [22], which is widely adopted to encode relations between sequence elements in text-to-SQL tasks, and which we use to build our two-stream encoder.

3.1 Task Formulation

The context-dependent text-to-SQL task is to generate the SQL query y_t given current user question u_t, historical question context $u_{<t} = \{u_1, u_2, \ldots, u_{t-1}\}$, and database schema $\mathcal{S} = \langle \mathcal{T}, \mathcal{C} \rangle$, which consists of a series of tables $\mathcal{T} = \{t_1, ..., t_{|\mathcal{T}|}\}$ and columns $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$.

3.2 Relation-Aware Transformer (RAT)

The relation-aware transformer is an extension of the vanilla transformer [21]. RAT can integrate the pre-defined relation features by adding relation embedding to the self-attention mechanism of the vanilla transformer:

$$e_{ij}^{(h)} = \frac{\mathbf{x}_i \mathbf{W}_Q^{(h)} \left(\mathbf{x}_j \mathbf{W}_K^{(h)} + \mathbf{r}_{ij}^K \right)^\top}{\sqrt{d_z / H}} \tag{1}$$

$$\mathbf{z}_i^{(h)} = \sum_{j=1}^{n} \alpha_{ij}^{(h)} \left(\mathbf{x}_j \mathbf{W}_V^{(h)} + \mathbf{r}_{ij}^V \right)^\top \tag{2}$$

where h denotes the h-th head, $a_{ij}^{(h)}$ is the attention weights, $\mathbf{W}_Q^{(h)}$, $\mathbf{W}_K^{(h)}$, $\mathbf{W}_V^{(h)}$ are learnable projection parameters, $\mathbf{r}_{ij}$ is the pre-defined relation embedding between input $\mathbf{x}_i$ and $\mathbf{x}_j$.

4 Methodology

Figure 3 illustrates our framework of QURG. It contains three parts: 1) Question rewriting model which is employed to obtain rewritten question u_t^{rw} from current question u_t and context $u_{<t}$. 2) Rewriting matrix generator which converts rewritten question to word-level relation matrix between u_t and $u_{<t}$. 3) SQL parser with two-stream encoder which can effectively integrate rewrite matrix for solving context dependencies.

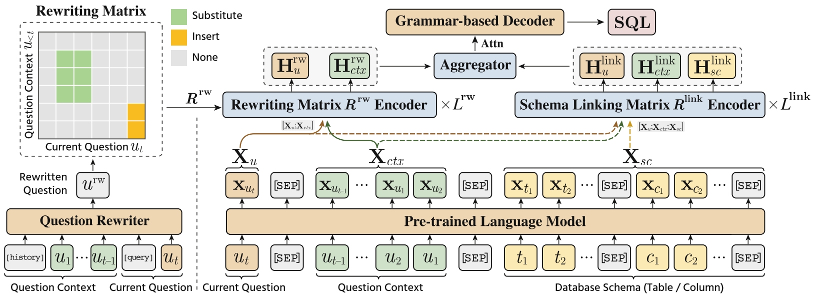


Fig. 3. Illustration of QURG framework: **Left:** Question rewriting module which produces rewritten questions u_t^{rw} and rewriting matrix R^{rw}. **Right:** PLM encoder and our proposed **two-stream relation matrix encoders**.

Table 1. Relation types between question and context. Relation edges exist from source token $x_i \in X_{\text{utter}}$ to target token $x_j \in X_{\text{utter}}$ if the pair meets one of the descriptions listed in the table, where X_{utter} is token set of u_t and $u_{<t}$.

Type of x_i	Type of x_j	Relation type	Description
Question u_t	Context $u_{<t}$	Q-C-Ins	Insert x_j before x_i
		Q-C-Sub	Substitute x_j for x_i
		None	None operation
Context $u_{<t}$	Question u_t	C-Q-Ins	Insert x_i before x_j
		C-Q-Sub	Substitute x_i for x_j
		None	None operation

4.1 Question Rewriting Model

Following [13] and [8], we employ a pre-trained T5-base sequence generator [5,17] as our QR model. Due to the lack of QR annotations on the text-to-SQL task, we directly use the *out-of-domain* QR dataset Canard [6] for QR model training and evaluation. Specifically, given the current user question u_t and historical context $u_{<t}$, we train QR models to produce rewritten question u_t^{rw} as: $\mathcal{P}(u_t^{\text{rw}}|\{\texttt{[history]}, u_{t-1}, \texttt{[query]}, u_t\})$, where [history] and [query] are special symbols to distinguish the context input and current question.

4.2 Rewriting Matrix Construction

Instead of feeding the rewritten question directly into the text-to-SQL model, we further convert the rewritten question into rewriting matrix which contains the key information to resolve context dependencies in current question. Following [16], we adopt a heuristic method to construct the bi-directional rewriting matrix. Bi-directional rewriting relation types between u_t and $u_{<t}$ are shown in Table 1.

Through the above method, we can associate the existing omission and co-reference in the current question u_t with the historical context $u_{<t}$, retaining context-dependencies in the form of a rewriting matrix, while ignoring trivial information or noise in the rewritten question u_t^{rw}.

4.3 QURG: SQL Parser with Rewriting Matrix

Our QURG model is an extension of RAT-SQL [22] following the common *encoder-decoder* architecture, which consists of three modules, as shown in Fig. 3: 1) Pre-trained Language Model (PLM) encoder which jointly transforms question u_t, context $u_{<t}$ and schema $\mathcal{S}$ into embedding as $\mathbf{X}_u, \mathbf{X}_{ctx}$ and $\mathbf{X}_{sc}$ respectively; 2) Two-stream relation matrix input encoder which further encodes element embedding with pre-defined pairwise relation features as $\mathbf{H}$; 3) Grammar-based decoder which generates SQL query corresponding to the current question.

Pre-trained Language Model Encoder. We concatenate the current question u_t, context $u_{<t}$ and schema $\mathcal{S}$ as the input sequence of pre-trained language models:

$$\begin{aligned} X = &\{\texttt{[CLS]}, u_t, \texttt{[SEP]}, u_{t-1}, ..., u_1, \texttt{[SEP]}, t_1, \\ &, t_2, ..., t_{|\mathcal{T}|}, \texttt{[SEP]}, c_1, c_2, ...c_{|\mathcal{C}|}, \texttt{[SEP]}\}. \end{aligned}$$

Following [2], we randomly shuffle the order of tables and columns in different mini-batches to alleviate the risk of over-fitting. Moreover, since each table name or column name may consist of multiple words, we use the average of the beginning and ending hidden vector as the schema element representation. Finally, the joint embedding vector of X is represented as $\mathbf{X} = \text{Concat}(\mathbf{X}_u; \mathbf{X}_{ctx}; \mathbf{X}_{sc})$.

Two-Stream Relation Matrix Encoder. This module contains two streams of relation matrix encoders: Schema Linking matrix R^{link} encoder and Rewriting matrix R^{rw} encoder. Firstly, the schema linking aids the model with aligning column/table references in the question and context to the corresponding schema columns/tables `id`. The schema linking relation matrix R^{link} is borrowed from RATSQL [22] which builds relations between natural language and schema elements. Through the schema linking method, we can get schema linking matrix $R^{\text{link}} \in \mathbb{R}^{(|X| \times |X|)}$. Then, the schema linking matrix R^{link} encoder takes joint embeddings of current question $\mathbf{X}_u$, context $\mathbf{X}_{ctx}$ and schema word $\mathbf{X}_{sc}$ as input and applies L^{link} stacked RAT layers to produce contextual representation $\mathbf{H}_u^{\text{link}}$, $\mathbf{H}_{ctx}^{\text{link}}$ and $\mathbf{H}_{sc}^{\text{link}}$ respectively:

$$\mathbf{H}_{(0)}^{\text{link}} = \text{Concat}\left(\mathbf{X}_u; \mathbf{X}_{ctx}; \mathbf{X}_{sc}\right) \tag{3}$$

$$\mathbf{H}_{(l)}^{\text{link}} = \text{RAT}_{(l)}\left(\mathbf{H}_{(l-1)}^{\text{link}}, R^{\text{link}}\right) \tag{4}$$

where $l \in [1, L^{\text{link}}]$ denote the index of the l-th RAT layer.

Table 2. Detailed statistics for SParC dataset [33] and CoSQL dataset [30].

Dataset	Question Interactions	Train/Dev/Test	Database/ Domain	User Questions	Average Turn	Vocab	System Response	Cross Domain
SParC	4,298	3,034/422/842	200/138	15,598	3.0	9,585	✗	✓
CoSQL	3,007	2,164/293/551	200/138	12,726	5.2	3,794	✓	✓

Similarly, the rewriting matrix R^{rw} encoder takes the joint embeddings of the current question $\mathbf{X}_u$ and context $\mathbf{X}_{ctx}$ as input and applies L^{rw} stacked RAT layers to get the rewriting enhanced representations $\mathbf{H}_u^{\mathrm{rw}}$, $\mathbf{H}_{ctx}^{\mathrm{rw}}$ of question and context respectively:

$$\mathbf{H}_{(0)}^{\mathrm{rw}} = \mathrm{Concat}\left(\mathbf{X}_u; \mathbf{X}_{ctx}\right) \tag{5}$$

$$\mathbf{H}_{(l)}^{\mathrm{rw}} = \mathrm{RAT}_{(l)}\left(\mathbf{H}_{(l-1)}^{\mathrm{rw}}, R^{\mathrm{rw}}\right) \tag{6}$$

where $l \in [1, L^{\mathrm{rw}}]$ denote the index of the l-th RAT layer.

Finally, we aggregate the representations of the two-stream encoder as:

$$\mathbf{H} = \mathrm{Concat}\left(\mathbf{H}_u^{\mathrm{link}} + \mathbf{H}_u^{\mathrm{rw}}; \mathbf{H}_{ctx}^{\mathrm{link}} + \mathbf{H}_{ctx}^{\mathrm{rw}}; \mathbf{H}_{sc}^{\mathrm{link}}\right)$$

Grammar-Based Decoder. We follow [22] and [2], using a grammar-based syntactic neural decoder that generates the target SQL action sequence in the depth-first-search order of the abstract syntax tree (AST). We refer the reader to [28] for details.

5 Experiments

In this section, we describe the experimental setups and evaluate the effectiveness of our proposed QURG. We compare QURG with previous works and conduct several ablation experiments. We also compare our method with the other two approaches of incorporating rewritten questions into text-to-SQL, to further verify the advantages of our QURG.

5.1 Experimental Setup

Datasets. We train our QURG model on two large-scale cross-domain context-dependent text-to-SQL datasets, SparC [33] and CoSQL [30]. The details of those datasets are organized in Table 2.

Evaluation Metrics. For evaluation, we employ two main metrics on both SParC and CoSQL datasets: *Question match* (**QM**) accuracy and *Interaction match* (**IM**) accuracy. Specifically, for **QM**, if all clauses in a predicted SQL are exactly matching those of the target SQL, the matching score is 1.0, otherwise, the score is 0.0. For **IM**, if all the predicted SQL in interaction is correct, the interaction match score is 1.0, otherwise the score is 0.0.

Implementation Details. For Text-to-SQL tasks, we use ELECTRA [4] as our pre-trained language model for all experiments. We set the learning rate to 1e-4, batch size to 32, and the maximum gradient norm to 10. The number of training epochs is 300 and 320 for SParC and CoSQL respectively. The numbers of RAT layers $L^{\text{link}} = 8$ for schema linking matrix encoder and $L^{\text{rw}} = 4$ for rewriting matrix encoder respectively. During inference, we set the beam size to 5 for SQL parsing.

5.2 Experimental Results

Table 3. Performances on the development set of SParC and CoSQL dataset. The models with $^\flat$ mark employ task adaptive pre-trained language models.

Models	SParC		CoSQL	
	QM	IM	QM	IM
EditSQL [34]	47.2	29.5	39.9	12.3
GAZP [36]	48.9	29.7	42.0	12.3
IGSQL [1]	50.7	32.5	44.1	15.8
RichContext [15]	52.6	29.9	41.0	14.0
IST-SQL [23]	47.6	29.9	44.4	14.7
R^{2}SQL [7]	54.1	35.2	45.7	19.5
DELTA [3]	58.6	35.6	51.7	21.5
SCoRE$^\flat$ [31]	62.2	42.5	52.1	22.0
HIE-SQL$^\flat$ [35]	64.7	45.0	56.4	**28.7**
RAT-SQL+TC$^\flat$ [12]	64.1	44.1	-	-
QURG (Ours)	**64.9**	**46.5**	**56.6**	26.6

Table 4. Detailed question match accuracy (**QM**) results in different interaction turns and SQL difficulties on the development set of SParC and CoSQL datasets. Results of ^{a} [34],^{b} [1],^{c} [23],^{d} [12],^{e} [23] and ^{f} [31] are from the original paper.

SParC (→)	Turn 1	Turn 2	Turn 3	Turn⩾ 4	/	Easy	Medium	Hard	Extra
EditSQL^a	62.2	45.1	36.1	19.3	/	68.8	40.6	26.9	12.8
IGSQL^b	63.2	50.8	39.0	26.1	/	70.9	45.4	29.0	18.8
R^{2}SQL^c	67.7	55.3	45.7	33.0	/	75.5	51.5	35.2	21.8
RAT-SQL+TC^d	**75.4**	64.0	**54.4**	40.9	/	-	-	-	-
QURG (Ours)	**75.4**	**66.1**	53.7	**44.3**	/	**80.1**	**64.4**	**43.4**	**35.1**
CoSQL (→)	Turn 1	Turn 2	Turn 3	Turn 4	Turn > 4	Easy	Medium	Hard	Extra
EditSQL^a	50.0	36.7	34.8	43.0	23.9	62.7	29.4	22.8	9.3
IGSQL^b	53.1	42.6	39.3	43.0	31.0	66.3	35.6	26.4	10.3
IST-SQL^e	56.2	41.0	41.0	41.2	26.8	66.0	36.2	27.8	10.3
SCoRE^f	60.8	53.0	47.5	49.1	32.4	-	-	-	-
QURG (Ours)	**64.5**	**55.4**	**55.7**	**50.0**	**42.3**	**77.2**	**50.0**	**40.5**	**20.6**

Table 5. Ablation studies for the components of QURG. Note that $-\text{Enc}_{\text{rw}}$ is also the baseline without the integration of rewritten questions $u_{<t}$.

Models	SParC		CoSQL	
	QM	**IM**	**QM**	**IM**
QURG	**64.9**	**46.5**	**56.6**	**26.6**
$-R^{\text{rw}}$	62.6	43.6	55.6	25.2
$-\text{Enc}^{\text{rw}}$	63.4	44.7	55.0	24.5

As shown in Table 3, we compare the performances of QURG with previous works on the development set of SParC and CoSQL datasets. QURG achieves comparable performance to previous state-of-the-art methods, including SCoRE [31], RAT-SQL-TC [12] and HIE-SQL [35] which effectively promote performances by using task-adaptive pre-trained language models. Besides, our QURG outperforms DELTA [3] which directly uses rewritten questions to predict SQL queries. In terms of **IM** accuracy on the CoSQL, QURG also surpasses PICARD [19] which is based on the super large pre-trained models T5-3B [17].

To further study the advantages of QURG on contextual understanding, as shown in Table 4, we evaluate the performances of the different question turns on SparC and CoSQL, and compare our QURG with previous powerful methods. Our QURG can achieve more improvements as the interaction turn increase. Furthermore, we evaluate the performance of QURG on the different difficulty levels of target SQL as shown in the right of Table 4, we observe that our QURG consistently achieves comparable performances to recent state-of-the-art works.

5.3 Ablation Study

As shown in Table 5, we conduct several ablation studies to evaluate the contribution of rewriting matrix integration for our QURG.

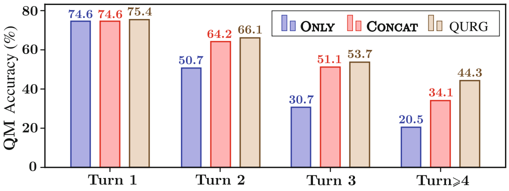


Fig. 4. Detailed results on different question turns for models ONLY, CONCAT and QURG.

To explore the effects of the rewriting matrix ($-R^{\text{rw}}$), we set all the relation types in the rewriting matrix to NONE and keep the model structure unchanged.

It degrades the model performances on both datasets by 1.0%–2.9% which confirms that rewriting matrix can effectively improve SQL parsing ability through enhanced context understanding. Then we further remove the whole rewriting matrix encoder ($-\text{Enc}^{\text{rw}}$) to verify the effect of the additional encoder parameters on question u_t and context $u_{<t}$, we observe that the additional parameters slightly degrade the performance on SParC, while slightly improving on CoSQL ($-\text{Enc}^{\text{rw}} \rightarrow -R^{\text{rw}}$), which indicates that the additional parameters have little effect on the improvements of QURG.

Table 6. Studies on different approaches to inject rewritten question into context-dependent text-to-SQL.

Models	SParC		CoSQL	
	QM	IM	QM	IM
ONLY	52.4	29.4	45.9	15.0
CONCAT	62.3	41.2	53.0	20.8
QURG	**64.9**	**46.5**	**56.6**	**26.6**

Moreover, we explore the effects of different approaches to integrating rewritten questions into text-to-SQL tasks, as shown in Table 6: 1) "**Only**" indicates only using rewritten questions u_t^{rw} to generate SQL queries, discarding the original question u_t and context $u_{<t}$; 2) "**Concat**" indicates concatenating original question u_t, context $u_{<t}$ with rewrite questions u_t^{rw}, treating u_t^{rw} as additional information. As shown in Table 6 and Fig. 4, ONLY feeding rewritten questions into text-to-SQL models results in a substantial performance drop, since the QR model is trained on *out-of-domain* data, the rewritten questions may contain a lot of noise. For "CONCAT", question and context are retained to meet the potential noise in rewritten questions, while it is still not ideal and causes performance to degraded against QURG, especially with the increase of turns, the performance of CONCAT decreases more significantly.

6 Conclusions

We propose QURG, a novel context-dependent text-to-SQL framework that utilizes question rewriting to resolve long-distance dependencies between the current question and historical context. Experimental results show that our QURG achieves comparable performance with recent state-of-the-art works.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62276017, U1636211, 61672081), the 2022 Tencent Big Travel Rhino-Bird Special Research Program, the 2022 CCF-NSFOCUS Kun-Peng Scientific Research Fund and the Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2021ZX-18).

References

1. Cai, Y., Wan, X.: IGSQL: database schema interaction graph based neural model for context-dependent text-to-SQL generation. In: Proceedings of EMNLP (2020)
2. 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 (2021)
3. Chen, Z., et al.: Decoupled dialogue modeling and semantic parsing for multi-turn text-to-SQL. In: Proceedings of ACL Findings (2021)
4. 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)
5. Dong, C., et al.: A survey of natural language generation. ACM Comput. Surv. (2023)
6. Elgohary, A., Peskov, D., Boyd-Graber, J.: Can you unpack that? Learning to rewrite questions-in-context. In: Proceedings of EMNLP (2019)
7. Hui, B., et al.: Dynamic hybrid relation exploration network for cross-domain context-dependent semantic parsing. In: Proceedings of AAAI (2021)
8. Kim, G., Kim, H., Park, J., Kang, J.: Learn to resolve conversational dependency: a consistency training framework for conversational question answering. In: ACL (2021)
9. Li, T., Fang, L., Lou, J.G., Li, Z.: TWT: table with written text for controlled data-to-text generation. In: Proceedings of EMNLP Findings (2021)
10. Li, T., Fang, L., Lou, J.G., Li, Z., Zhang, D.: Anasearch: extract, retrieve and visualize structured results from unstructured text for analytical queries. In: Proceedings of WSDM (2021)
11. Li, Y., et al.: On the (in)effectiveness of large language models for Chinese text correction. CoRR (2023)
12. Li, Y., Zhang, H., Li, Y., Wang, S., Wu, W., Zhang, Y.: Pay more attention to history: a context modeling strategy for conversational text-to-SQL. arXiv:2112.08735 (2021)
13. Lin, S.C., Yang, J.H., Nogueira, R., Tsai, M.F., Wang, C.J., Lin, J.: Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. arXiv preprint arXiv:2004.01909 (2020)
14. Lin, X.V., Socher, R., Xiong, C.: Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In: Proceedings of EMNLP Findings (2020)
15. Liu, Q., Chen, B., Guo, J., Lou, J.G., Zhou, B., Zhang, D.: How far are we from effective context modeling? An exploratory study on semantic parsing in context. In: Proceedings of IJCAI (2020)
16. Liu, Q., Chen, B., Lou, J.G., Zhou, B., Zhang, D.: Incomplete utterance rewriting as semantic segmentation. In: Proceedings of EMNLP (2020)
17. 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)
18. Scholak, T., Li, R., Bahdanau, D., de Vries, H., Pal, C.: Duorat: towards simpler text-to-SQL models. In: Proceedings of AACL (2021)
19. Scholak, T., Schucher, N., Bahdanau, D.: PICARD: parsing incrementally for constrained auto-regressive decoding from language models. In: Proceedings of EMNLP (2021)
20. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of AACL (2018)
21. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)

22. 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 (2020)
23. Wang, R.Z., Ling, Z.H., Zhou, J., Hu, Y.: Tracking interaction states for multi-turn text-to-SQL semantic parsing. In: Proceedings of AAAI (2021)
24. Yang, J., et al.: Multilingual machine translation systems from microsoft for WMT21 shared task. In: Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, 10–11 November 2021 (2021)
25. Yang, J., Ma, S., Zhang, D., Wu, S., Li, Z., Zhou, M.: Alternating language modeling for cross-lingual pre-training. In: Proceedings of AAAI (2020)
26. Yang, J., et al.: Learning to select relevant knowledge for neural machine translation. In: Proceedings of NLPCC (2021)
27. Yang, J., et al.: Multilingual agreement for multilingual neural machine translation. In: Proceedings of ACL (2021)
28. Yin, P., Neubig, G.: TRANX: a transition-based neural abstract syntax parser for semantic parsing and code generation. In: Proceedings of EMNLP (2018)
29. Yu, T., et al.: GraPPa: grammar-augmented pre-training for table semantic parsing. In: Proceedings of ICLR (2021)
30. Yu, T., et al.: CoSQL: a conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In: Proceedings of EMNLP (2019)
31. 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 (2021)
32. 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 (2018)
33. Yu, T., et al.: SParC: cross-domain semantic parsing in context. In: Proceedings of ACL (2019)
34. Zhang, R., et al.: Editing-based SQL query generation for cross-domain context-dependent questions. In: Proceedings of EMNLP (2019)
35. 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 Findings (2022)
36. Zhong, V., Lewis, M., Wang, S.I., Zettlemoyer, L.: Grounded adaptation for zero-shot executable semantic parsing. In: Proceedings of EMNLP (2020)

Sarcasm Relation to Time: Sarcasm Detection with Temporal Features and Deep Learning

Md Saifullah Razali[1,2](✉), Alfian Abdul Halin[1](✉), Yang-Wai Chow[2](✉), Noris Mohd Norowi[1](✉), and Shyamala Doraisamy[1](✉)

[1] Universiti Putra Malaysia, Jalan Universiti 1, 43400 Serdang, Selangor, Malaysia
saifullahbjr8500@gmail.com, {alfian,noris,shyamala}@upm.edu.my
[2] University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia
caseyc@uow.edu.au

Abstract. This paper presents a deep learning-based framework to detect sarcasm in relation to time. Deep N-gram features generated using the FastText algorithm, combined with temporal handcrafted temporal features are used to train several machine learning classifiers. Experimental results show that Logistic Regression performs the best among all the classifiers. The introduction of the handcrafted temporal features has also whosn to improve overall detection performance when compared to existing works in the field.

Keywords: Sarcasm Detection · Temporal Features · Natural Language Processing · Deep Learning · Sentiment Analysis

1 Introduction

The multitude of personal social media data generated up till today has shown to be useful for analysis purposes, where organizations can use this data to better understand their target audience [4,17]. Specifically, the main interest is to understand sentiment, hence the growing field of sentiment analysis [24].

Sarcasm can be defined as a positive sentence (or phrase) with negative intent [33]. In Natural Language Processing (NLP), the ability for systems to automatically identify and detect sarcasm, even when using state-of-the-art algorithms, is still very challenging [25]. Wrongly identifying sarcasm (or vice versa) can change the polarity of a sentence and therefore jeopardize overall sentiment analysis [20,25]. Earlier works such as [12,33] attempted sarcasm detection using rule-based techniques. This quickly became problematic as the number of rules to consider became bigger and bigger. Hence, more recent studies such as [26,30] began to adopt machine learning and deep learning where algorithms could be trained either on engineered (handcrafted) features or through automatic feature discovery.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 287–297, 2024.
https://doi.org/10.1007/978-981-99-7022-3_25

In this paper, both deep and machine learning are applied. Deep features are generated through a deep learning architecture. These are then combined with handcrafted temporal features to represent each data instance (tweet). The significant contribution in this work is the use of temporal features, which attempts to represent different tones and gestures used in normal day-to-day conversations. This is based on the intuition that people tweeting sarcasm commonly add 'creativity' to their tweets. These come in the form of semantic clues indicating their sarcasm [12,33]. These clues is what this work is trying to represent and manipulate, particularly the temporal clues. All features are generated from the dataset found in [31].

2 Related Work

Earlier works in sarcasm detection employed rule-sets. Barbieri et al. [7] identifies sarcasm by looking at frequency of words. Their technique is slightly advanced by [11] by adding more rules in the form of word patterns. Among features taken into consideration in the rules are the number of +ve/-ve words, and the number of words based on emotion. Rules are also used in [35] where the authors define specific seed phrases. These are such as "being sarcastic", which are then to identify other sarcastic instances.

A majority of sarcasm detection work employ deep learning architectures, such as [30] and [26]). For example, [30] designed their own Convolutional Neural Network (CNN) to discover feature sets, which in turn are used train a Support Vector Machine (SVM) classifier. The work in [19] based their detector on Embeddings from Language Models (ELMo), which derives its vector representation from a bi-directional Long Short Term Memory (for the fundamentals of this technique, readers can be directed to [29]. Riloff et al. [33] used classifiers to locate positive verbs that occur together in negative situations, followed by other researches to further refine their own rule-sets (e.g. [6,21,30])

Historical user tweets were also used for feature generation. The authors in [6] analysed the thread of tweets, from the same user, for relevance. Relevance is indicated when features exhibit specific rules. For example, "audience feature" is evident in historical correspondences between the tweet's author and its intended recipient. Kreuz & Caucci [22] furthered this idea based on their claim that sarcasm is more commonly directed/projected towards people whom the writer knows (or is familiar with). Other works that follow this trend are [3,16,32]. For example, Rajadesingan et al. [32] proposed their SCUBA (Sarcasm Classification Using a Behavioural Modeling Approach) framework. SCUBA classifies a user's behaviour based on Bamman & Smith [6]'s method, but additionally considered the difference between current and past tweets.

In this paper, a deep feature set (in the form of N-grams) is used along with a handcrafted set (i.e. temporal set). The justification for this temporal set is based on the results of [13], where it is strongly assumed that sarcastic sentences can have a negative connotation at one instance of time, but can change into a positive at a different instance of time. In addition, when a person is being sarcastic, abnormal tones are commonly utilized [5,34] as well as exaggerations as stated in [23]. Our feature set is based on these assumptions.

3 Proposed Method

The overall work flow of our work is shown in Fig. 1.

3.1 Data Acquisition

For our experiments, the publicly available dataset in [31] is used. It is worth noting that this dataset is imbalanced, where from the total of 780,000 English tweets, 130,000 are sarcastic where 650,000 are non-sarcastic. The authors however justify this imbalance as sarcastic utterances occur more rarely. In this work, 80 percent is used for training whereas 20 percent is used for testing.

3.2 Data Preprocessing

We follow the standards set by [16], since preprocessing is essential in any Natural Language Processing (NLP) task. The five types of preprocessing techniques applied are (i) conversion to lower-case, (ii) punctuation removal, (iii) "#sarcasm" hashtags removal, (iv) stopwords removal, and (v) lemmatization.

3.3 Sarcasm Detection

Deep Learning Extraction. As mention, the majority of work in sarcasm detection is mostly done using deep learning (DL) [16,26,30,36]. The biggest advantage of DL is its ability to generate optimal features for various NLP tasks [26,30].

This endeavor involves constructing a Convolutional Neural Network architecture to derive ten profound features. The specific architecture for sarcasm detection task is shown in Fig. 2.

FastText. Notice in Fig. 2 that before going into the CNN, word-embedding is firstly performed. We adopt FastText [10] to convert tweets into the CNN input vectors. Note that we choose FastText as compared to the commonly used Word2Vec [27] because our input features are N-grams (as opposed to individual words generated by Word2Vec). For example, FastText breaks the word "human" into the trigram "hum", "uma" and "man". Note also that FastText has the ability to generate unigrams and birgams as well. FastText's output is therefore a word-embedding vector of all broken N-grams in the dataset. We see this as giving a more effective representation for each data instance, especially for rare and/or wrongly spelled words.

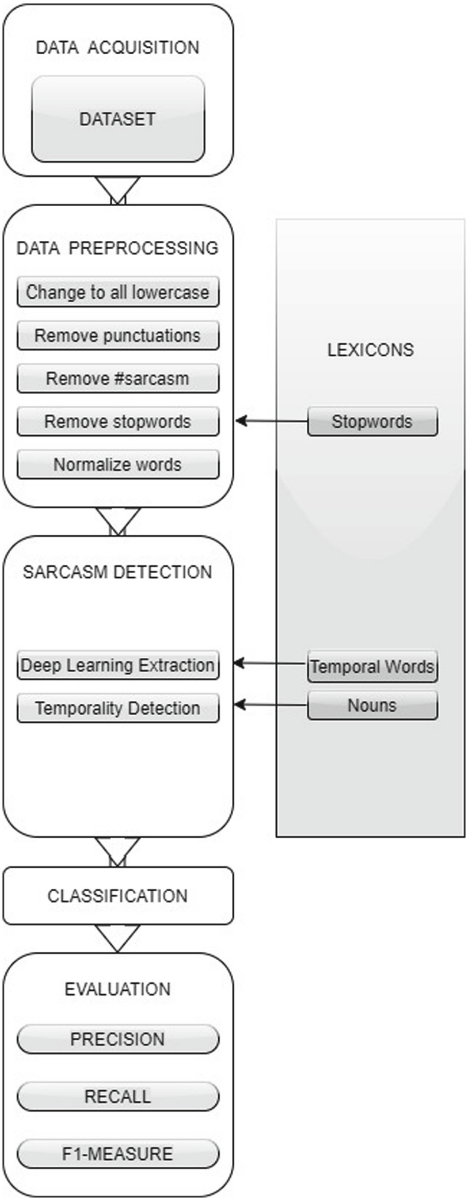


Fig. 1. Overall Framework of the Study

Convolutional Neural Network (CNN): The deep features extraction part is performed by out vanilla CNN architecture (Fig. 3). This is the detail of the CNN part from the overall architecture of the sarcasm detection part shown in Fig. 2.

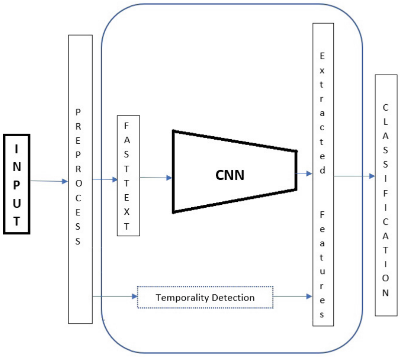


Fig. 2. Overall Architecture of Sarcasm Detection Part

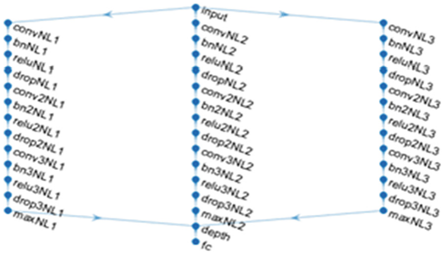


Fig. 3. Vanilla CNN for detecting sarcasm

The CNN in Fig. 3 is displayed in a top-down fashion. Note that *NL* is the abbreviation for *N*-gram Length. The specifics for the CNN is as follows:

1. The input layer: $1 \times 100 \times N$, where N is the number of instances from the dataset. Embedded-word vectors are the initial input.
2. Layer before concatenation:
   - 1 convolutional layer consisting 200-neurons (filter size $= 1 \times 100 \times N$, with a stride of 1), followed by
   - 2 convolutional layers consisting 200-neurons (filter size $= 1 \times 100 \times 200$, with a stride of 1), followed by
   - 3 batch normalizations (200-channels), followed by
   - 3 Rectified Linear Unit (ReLU) activations, followed by
   - 3 dropout layers (20%), followed by
   - 1 max-pooling layer with stride 1, followed by
3. A depth concatenation layer
4. Finally, a fully connected layer of 10-neurons.

Specifically, the convolutional layers will produce feature maps followed by batch normalizations that also improve training time and stability. Regularization is performed at a minimum of 20% dropouts so that overfitting does not occur. The inputs are in the form of word vectors generated by Fasttext, where the vector size is set to [1 100]. These vectors are respectively split into their own groups of N-grams, i.e. - unigrams, bigrams, and trigrams. These groups are then fed into the three concurrent graph architectures. The deep features vector are the 10-neuron fully connected layer at the end. These are finally concatenated with the handcrafted temporal features for final instances representation.

3.4 Temporality Detection

According to [13], sarcasm also has to do with time (hence, the temporal dimension). For example, the following phrase "You think like Trump!" if uttered in 2016, after Donald Trump won the United States elections, would have a different connotation then if it were uttered in the year 2020 (after he was ousted). This illustrates how time plays a role in determining whether sarcasm is implied or otherwise.

Due to this, we identify temporal features. These features are handcrafted where specific rules apply depending on the lexicon. These lexicons are explained as follows.

3.5 Lexicons

All the handcrafted features are extracted using the following two lexicons:

1. **Temporal Words:** These are more commonly referred to as transition words in grammar. These words relate to time and examples are such as *previously*, *instead* and *furthermore*. For a more complete list, readers can be directed to [2]. In this work, a list of 52-instances were used.
2. **Nouns:** This lexicon is downloaded from [1]. A total of 1500-instances are used.

3.6 Experimental Results

We evaluated the performance of five different *traditional* machine learning classifier based on the handcrafted and deep features combination. These are the (i) K-Nearest Neighbor (KNN) [15], (ii) Support Vector Machine (SVM) [28], (iii) Decision Tree (DT) [8], (iv) Logistic Regression (LR) [9], and (v) Discriminant Analysis (DISCR) [14].

For each classifier, we obtained the following metrics (where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives):

Accuracy:

$$accuracy = \frac{Correct\ predictions}{Total\ predictions} \tag{1}$$

Precision:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

Recall:

$$recall = \frac{TP}{TP + FN} \tag{3}$$

F1-measure:

$$F1 = 2.\ \frac{precision\ \cdot\ recall}{precision + recall} \tag{4}$$

It is common to determine a classification algorithm's *accuracy* score to determine effectiveness [18]. However, since this measure is an overall view of how the classifier performs on both positive and negative classes, it might not provide the best insight. Hence, *precision* and *recall* are also calculated as they provide insights on classifier performance with regards to each class (i.e. *sarcastic* vs. *non-sarcastic*). Specifically, *precision* looks at the proportion of genuinely sarcastic instances out of all the system-predicted sarcastic instances. When this ratio is low, only a small quantity of truly sarcastic tweets are identified as such. *Recall* on the other hand looks at actual sarcastic instances over everything that is actually sarcastic. When this measure is low, many sarcastic tweets are classified as non-sarcastic. $F1 - measure$, also called the harmonic mean, attempts to provide an overall measure of precision and recall. This is especially useful when dealing with imbalanced datasets.

For the experiments, we divided the dataset [31] into a 64 : 16 : 20 split for training, validation and testing, respectively. Specifically, from the overall 780,000 English language tweets, 83% are non-sarcastic instances, whereas the remaining 17% are sarcastic instances.

4 Results and Discussion

4.1 Classification Results

Table 1 shows the metrics calculated for each classifier, based on the handcrafted and deep features combination.

Table 1. Classification Results for the SVM, KNN, LR, DT and DISCR

Classification Algorithm	F1	Precision	Recall	Accuracy
SVM	0.87	0.87	0.87	87%
KNN	0.86	0.86	0.86	86%
LR	**0.89**	**0.90**	**0.89**	**89%**
DT	0.87	0.87	0.87	87%
DISCR	0.87	0.87	0.86	86%

The Logistic Regression (LR) classifier seems to be giving the overall superior results, based on all calculated metrics. The SVM and Decision Tree on the other hand are also promising, showing high accuracy and F1-measure. Since LR performed the best, further analysis in this paper will be based on its results.

4.2 Logistic Regression vs. Existing Works

Using the same dataset, we compared the LR classifier with works from Ilić et al. [19] and Shmueli et al. [35], as to the best of our knowledge, these works are most relevant. The comparative results are shown in Table 2.

Table 2. Performance comparison with Existing Works

Method	F1	Precision	Recall	Accuracy
Ilić et al. [19]	0.87	0.87	0.87	88%
Shmueli et al. [35]	0.86	0.87	**0.91**	87%
Proposed Method	**0.89**	**0.90**	0.89	**89%**

Overall, our proposed framework shows better performance across almost metrics (*recall* is slightly lower at 0.89 but arguably comparable).

4.3 Performance Comparison Among Feature Sets

The performance of feature sets are also presented. For this, we only measure *accuracy* (Fig. 4).

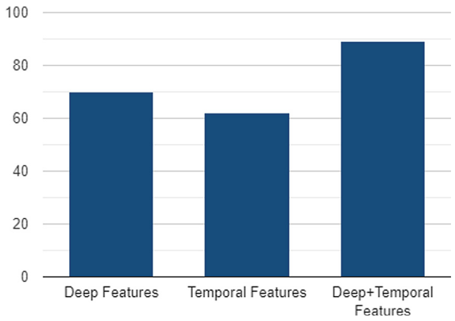


Fig. 4. Performance of Individual Feature Sets

Compared to deep feature, temporal feature shows a lower performance. Based on our observations, the main reason for this is not frequently present in the dataset. We also observed that, since informal language is most often used on Twitter, the detection of temporal words become more difficult to reliably detect. However, we argue that an accuracy of more than 60% shows the importance of this feature and that it adds value to the overall performance of a sarcasm detection system.

5 Conclusion Remarks and Future Works

The results provide insights into the use of handcrafted temporal features and deep features for sarcasm detection. The temporal features seem to provide added value for tweets that contain temporal/transitional words. Experimental results have demonstrated that when the temporal features are combine with deep features, with a Logistic Regression classifier as the main classifier, all $presision$, $recall$, $F1 - measure$ and overall $accuracy$ is high. In the future, more datasets will be explored with the hope of better generalization.

References

1. Noun. https://www.talkenglish.com/vocabulary/top-1500-nouns.aspx. Accessed 21 Feb 2021
2. Temporal words. https://grammar.yourdictionary.com/style-and-usage/list-transition-words.html. Accessed 21 Feb 2021
3. Amir, S., Wallace, B.C., Lyu, H., Silva, P.C.M.J.: Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976 (2016)
4. Aquino, J.: Transforming social media data into predictive analytics. CRM Mag. **16**(11), 38–42 (2012)
5. Attardo, S., Eisterhold, J., Hay, J., Poggi, I.: Multimodal markers of irony and sarcasm. Humor **16**(2), 243–260 (2003)
6. Bamman, D., Smith, N.A.: Contextualized sarcasm detection on twitter. In: Ninth International AAAI Conference on Web and Social Media. Citeseer (2015)
7. Barbieri, F., Saggion, H., Ronzano, F.: Modelling sarcasm in twitter, a novel approach. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 50–58 (2014)
8. Belson, W.A.: A technique for studying the effects of a television broadcast. J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) **5**(3), 195–202 (1956)
9. Berkson, J.: A statistically precise and relatively simple method of estimating the bio-assay with quantal response, based on the logistic function. J. Am. Stat. Assoc. **48**(263), 565–599 (1953)
10. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
11. Bouazizi, M., Ohtsuki, T.O.: A pattern-based approach for sarcasm detection on twitter. IEEE Access **4**, 5477–5488 (2016)
12. Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcasm in Twitter and Amazon. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp. 107–116 (2010)
13. Ebrahimi, M., Yazdavar, A.H., Sheth, A.: Challenges of sentiment analysis for dynamic events. IEEE Intell. Syst. **32**(5), 70–75 (2017)
14. Fisher, R.A.: The statistical utilization of multiple measurements. Ann. Eugenics **8**(4), 376–386 (1938)
15. Fix, E., Hodges, J.L.: Discriminatory analysis: nonparametric discrimination: small sample performance (1952)

16. Ghosh, A., Veale, T.: Fracking sarcasm using neural network. In: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 161–169 (2016)
17. Ghosh, A., Veale, T.: Magnets for sarcasm: making sarcasm detection timely, contextual and very personal. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 482–491 (2017)
18. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. Knowl. Data Eng. **17**(3), 299–310 (2005)
19. Ilić, S., Marrese-Taylor, E., Balazs, J.A., Matsuo, Y.: Deep contextualized word representations for detecting sarcasm and irony. arXiv preprint arXiv:1809.09795 (2018)
20. Joshi, A., Agrawal, S., Bhattacharyya, P., Carman, M.J.: *Expect the unexpected*: harnessing sentence completion for sarcasm detection. In: Hasida, K., Pa, W.P. (eds.) PACLING 2017. CCIS, vol. 781, pp. 275–287. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8438-6_22
21. Joshi, A., Bhattacharyya, P., Carman, M.J.: Automatic sarcasm detection: a survey. ACM Comput. Surv. (CSUR) **50**(5), 1–22 (2017)
22. Kreuz, R., Caucci, G.: Lexical influences on the perception of sarcasm. In: Proceedings of the Workshop on Computational Approaches to Figurative Language, pp. 1–4 (2007)
23. Kunneman, F., Liebrecht, C., Van Mulken, M., Van den Bosch, A.: Signaling sarcasm: from hyperbole to hashtag. Inf. Process. Manage. **51**(4), 500–509 (2015)
24. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
25. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C., Zhai, C. (eds.) Mining Text Data, pp. 415–463. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3223-4_13
26. Majumder, N., Poria, S., Peng, H., Chhaya, N., Cambria, E., Gelbukh, A.: Sentiment and sarcasm classification with multitask learning. IEEE Intell. Syst. **34**(3), 38–43 (2019)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
28. Müller, K.-R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Predicting time series with support vector machines. In: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 999–1004. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0020283
29. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
30. Poria, S., Cambria, E., Hazarika, D., Vij, P.: A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815 (2016)
31. Ptáček, T., Habernal, I., Hong, J.: Sarcasm detection on Czech and English twitter. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 213–223 (2014)
32. Rajadesingan, A., Zafarani, R., Liu, H.: Sarcasm detection on twitter: a behavioral modeling approach. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 97–106 (2015)
33. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 704–714 (2013)

34. Rockwell, P.: Vocal features of conversational sarcasm: a comparison of methods. J. Psycholinguist. Res. **36**(5), 361–369 (2007)
35. Shmueli, B., Ku, L.-W., Ray, S.: Reactive supervision: a new method for collecting sarcasm data. arXiv preprint arXiv:2009.13080 (2020)
36. Zhang, M., Zhang, Y., Fu, G.: Tweet sarcasm detection using deep neural network. In: Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers, pp. 2449–2460 (2016)

Self-agreement: A Framework for Fine-Tuning Language Models to Find Agreement Among Diverse Opinions

Shiyao Ding(✉) and Takayuki Ito

Kyoto University, Kyoto-shi, Kyoto 606-8501, Japan
{ding,ito}@i.kyoto-u.ac.jp

Abstract. Finding an agreement among diverse opinions is a challenging topic in social intelligence. Recently, large language models (LLMs) have shown great potential in addressing this challenge due to their remarkable capabilities in comprehending human opinions and generating human-like text. However, they typically rely on extensive human-annotated data. In this paper, we propose Self-Agreement, a novel framework for fine-tuning LLMs to autonomously find agreement using data generated by LLM itself. Specifically, our approach employs the generative pre-trained transformer-3 (GPT-3) to generate multiple opinions for each question in a question dataset and create several agreement candidates among these opinions. Then, a bidirectional encoder representations from transformers (BERT)-based model evaluates the agreement score of each agreement candidate and selects the one with the highest agreement score. This process yields a dataset of question-opinion-agreements, which we use to fine-tune a pre-trained LLM for discovering agreements among diverse opinions. Remarkably, a pre-trained LLM fine-tuned by our Self-Agreement framework achieves comparable performance to GPT-3 with only 1/25 of its parameters, showcasing its ability to identify agreement among various opinions without the need for human-annotated data.

Keywords: Social intelligence · Consensus building · Opinion summarization · Large language models

1 Introduction

Social intelligence (SI) is the capacity to navigate relationships effectively and garner cooperation from others [2,4]. A crucial component of SI is consensus building, which involves achieving substantial agreement within a group on a particular topic. This studies on how to reach an agreement through discussions involving a variety of opinions, which is often challenging [4]. The recent rise and success of large language models (LLMs), such as the generative pre-trained transformer-3 (GPT-3) [3], offer promising opportunities for addressing

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 298–309, 2024.
https://doi.org/10.1007/978-981-99-7022-3_26

this challenge, by leveraging their capabilities in comprehending human opinions and generating human-like text.

Currently, employing LLMs for finding agreement among diverse opinions can be divided into two kinds of approaches. The first approach involves zero-shot learning on a pretrained LLM, such as GPT-3/4, which can output agreement without additional training. However, accessing these models requires the use of an application programming interface (API), which can introduce latency, high costs, and limited accessibility for research. Moreover, while some pretrained LLMs like Meta's LLaMA [9] are available for local download, their massive parameters often lead to substantial storage and computational requirements during inference, making them less feasible for certain applications.

The second approach focuses on few-shot learning, which entails fine-tuning a LLM using a dataset containing opinions and agreement data. One notable example is [1], which relies on a 70 billion parameter model and expert-annotated data. However, this dependence on high-quality, human-generated data and the associated costs can be a significant barrier for many researchers and organizations seeking to leverage these models for consensus-building tasks. These limitations highlight the need for more accessible and cost-effective solutions in applying LLMs to find agreement among diverse opinions without heavily relying on human resources.

In this paper, we address the above challenges by proposing Self-Agreement, a framework designed to autonomously find agreement among diverse opinions using LLMs. Our method consists of the following four key steps. First, we employ GPT-3 to generate multiple opinions on each question given a dataset containing various questions. Next, we use GPT-3 to generate several potential agreements among these diverse opinions for each question. We then evaluate the agreement score of each agreement candidate, choosing the one with the highest score as the optimal agreement to construct a question-opinion-agreement dataset. Finally, we fine-tune a pre-trained LLM using the above dataset. This step allows the model to adapt to the specific task of finding agreements among various opinions. In our evaluation, we fine-tuned a 7 billion pre-trained LLM using our Self-Agreement framework. Remarkably, our model achieves a similar performance to GPT-3 with only 1/25 of its parameters, showcasing its ability to identify areas of agreement among conflicting views.

In summary, our contributions include: 1) a large dataset consisting of various questions, opinions, and agreement candidates, which can serve as a valuable resource for building and evaluating consensus-building models; 2) the Self-Agreement framework which is designed to autonomously find agreement among diverse opinions using LLMs without human-annotated data; 3) a demonstration of fine-tuning a pre-trained model, with a comprehensive set of experiments confirming its effectiveness and performance.

2 Related Work

The concept of agreements can vary depending on the task. In this paper, we consider agreement as a form of opinion summarization from users, which may be agreed upon by all the users, similar to the approaches used in [1,8].

Opinion Summarization. Most opinion summarization methods follow a three-step process [5]: First, *aspect extraction* is performed, which involves identifying the relevant features or aspects of the product that the user is commenting on. Second, *sentiment prediction* is used to determine the sentiment of the extracted aspects, whether it is positive, negative, or neutral. This step helps to understand the overall opinion of the user. Finally, *summary generation* is used to present the identified opinions to the user in a concise and easily understandable manner. This step involves condensing the extracted aspects and their corresponding sentiments into a brief summary. Most methods rely on extractive techniques for creating textual summaries, which select representative segments from the source text. However, this can result in loss of information that may be useful depending on user needs.

Most current approaches for opinion summarization, as described in the reference [7], involve encoding documents and then decoding the learned representations into an abstractive summary. These methods leverage the success of sequence-to-sequence neural network architectures and are trained using sets of opinions and their corresponding summaries. In this approach, there is no need to explicitly identify aspects and sentiment for the opinion summarization task, as these are learned implicitly from the training data. However, due to memory limitations, training these models end-to-end with a large number of input reviews for each target entity is practically infeasible [6].

LLMs for Opinion Summarization. The LLMs have shown great potential in opinion summarization tasks, largely due to their ability to process and generate natural language. Bakker et al. [1] fine-tuned a 70 billion parameter pretrained LLM to produce statements that maximize the expected approval for groups with diverse opinions. The model demonstrated exceptional performance, generating consensus statements preferred by over 70% of human users compared to prompted LLMs. However, their approach relies on human annotations. Although the Self-Instruction framework proposed by Wang et al. [10] utilizes pre-trained LLM outputs for fine-tuning LLMs to follow human instructions, it still requires the preparation of seed instructions. Our Self-Agreement framework, on the other hand, specifically targets consensus-building tasks and does not depend on any human-generated instructions. This distinction highlights the advantages of the Self-Agreement framework in handling diverse opinions and generating agreement statements without extensive reliance on human-annotated data.

3 Problem and Method

3.1 Consensus-Building Problem

In this section, we address the problem of identifying agreement among a set of users, represented as $N = \{1, ..., j, ..., |N|\}$. We first define a consensus-building instance CB^i for each question i described by the tuple $CB^i =< q^i, OP^i, AC^i >$. Here, q^i is the question i, $OP^i = \{op_1^i, ..., op_j^i, ..., op_{|N|}^i\}$ represents the set of opinions (with op_j^i denoting user j's opinion for question q^i), and $AC^i = \{ac_1^i, ..., ac_{|AC^i|}^i\}$ is set of potential agreement candidates. Agreement, as previously outlined, is understood as an opinion summary that could be universally accepted by the user base. For each user j, there exists a function $Prob_j : OP^i \times AC^i \rightarrow [0, 1]$ which determines the likelihood of agreement with an opinion $op_j^i \in OP^i$ based on a specific agreement candidate $ac_k^i \in AC^i$. For instance, $\mathrm{Prob}_j(op_j^i, ac_k^i) = 0.6$ indicates a 60% probability that user j would agree with the agreement candidate ac_k^i. An agreement threshold ϵ^i is defined for each question, establishing agreement if all users possess a probability exceeding this threshold, i.e.,

$$\forall j \in N, Prob_j > \epsilon^i \tag{1}$$

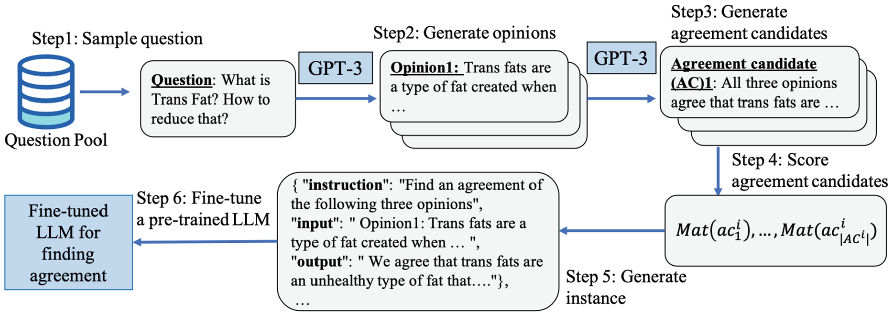

Step1: Sample question
Question Pool
Question: What is Trans Fat? How to reduce that?
GPT-3
Step2: Generate opinions
Opinion1: Trans fats are a type of fat created when …
GPT-3
Step3: Generate agreement candidates
Agreement candidate (AC)1: All three opinions agree that trans fats are …
Step 4: Score agreement candidates
$Mat(ac_1^i), ..., Mat(ac_{|AC^i|}^i)$
Step 5: Generate instance
{ "instruction": "Find an agreement of the following three opinions",
"input": " Opinion1: Trans fats are a type of fat created when … ",
"output": " We agree that trans fats are an unhealthy type of fat that…."},
…
Step 6: Fine-tune a pre-trained LLM
Fine-tuned LLM for finding agreement

Fig. 1. The framework of the Self-Agreement.

3.2 Automatic Opinion-Agreement Data Generation

Generating agreement data mentioned in 3.1 for consensus building often entails significant human involvement, especially when dealing with vast arrays of human-crafted questions and corresponding agreement data. In this section, we harness the capabilities of LLMs to automate the creation of consensus-building instances, streamlining the entire process as depicted in Fig. 1.

For each question q^i, our strategy for generating opinions leans on the principles of the Self-Agreement method. This method capitalizes on the prowess of extensively pre-trained language models, such as GPT-3, prompting them to forge new and diverse instructions in a cyclical bootstrapping manner. Specifically, our model employs GPT-3 to produce a range of diverse opinions using the prompt “Generate $|OP^i|$ opinions for the question q^i,” illustrated in Steps 1 and 2 of Fig. 1. A distinction worth noting is that while the Self-Instruct approach hinges on an initial batch of human-crafted opinions as seeds, our Self-Agreement method is devoid of any reliance on pre-existing human opinions.

Proceeding to Step 3, concerning the generation of agreement, GPT-3 is once again put to task. This time, it is to conceive $|AC^i|$ agreement candidates in response to the prompt “Find an agreement for the following opinions OP^i,” with a representative example presented in Table 1. Given a collection $Q = \{q^1, q^2, ..., q^{|Q|}\}$ of questions, our overarching goal is to fabricate $|Q|$ consensus-building instances, culminating in a cohesive question-opinion-agreement dataset $\{CB^1, CB^2, ..., CB^{|Q|}\}$.

3.3 Scoring Agreement Candidates

In every consensus-building instance, the objective is to identify an agreement as delineated in Eq. (1). However, directly ascertaining $Prob^j$ for each user is challenging. To address this, we introduce a scoring function $Mat : OP^i \times AC^i \rightarrow \mathcal{R}$ to gauge the compatibility between agreement ac^i_k and opinion op^i_j.

In this work, Mat is bounded within the interval [0, 1]. Specifically, $Mat(op^i_j, ac^i_k) = 0$ indicates that the agreement ac^i_k is entirely unrelated to opinion op^i_j, while a score of 1 implies complete alignment. As an illustration, when ac^i_k matches op^i_j, the score is unity: $Mat(op^i_j, ac^i_k)|_{ac^i_k = op^i_j} = 1$.

For each question q^i, with $|OP^i|$ opinions and $|AC^i|$ agreement candidates, the model assigns a score reflecting each candidate agreement’s pertinence and congruence with the opinions. By aggregating scores across all agreement candidates, we identify the preeminent candidate encapsulating diverse perspectives. The ideal agreement candidate ac^i_* boasts the apex agreement score:

$$ac^i_* = \mathrm{argmax}_{ac^i_k \in AC^i} \sum_{op^i_j \in OP^i} Mat(op^i_j, ac^i_k) \tag{2}$$

While a myriad of functions can serve as the scoring mechanism, this study leverages a BERT-based model to assess the semantic congruence between opinions and agreements. This model extracts contextualized embeddings for sentences, encapsulating their nuanced meanings. A similarity metric, like cosine similarity, then evaluates the proximity of these embeddings. Thus, the result of an instance is given by ac^i_*, as expounded in Step 5 of Fig. 1.

3.4 Fine-Tuning the Language Model

After building the question-opinion-agreement dataset, we fine-tune a pre-trained language model for the task of consensus building, as shown in Step 6 of Fig. 1. In this paper, we employ a 7-billion(7B)-parameter LLaMA model based on the Alpaca-LoRA architecture for fine-tuning. The Alpaca-LoRA architecture comprises two components: 1) the original 7-billion-parameter model,

Table 1. A sample of self-agreement process with three opinions and two candidate agreements on the topic of trans fats.

Topic	How the human species evolved?
Opinion 1	Humans evolved from the African continent, where the climate and environment provided the perfect conditions for the species to develop and thrive.
Opinion 2	Humans evolved through the process of natural selection, where traits that allowed for greater survival and reproduction were selected for, and those that did not were eliminated.
Opinion 3	Humans evolved through the process of genetic mutation, where random changes in the genetic code allowed for new traits and adaptations to develop.
Agreement Candidate 1	All three opinions agree that humans evolved through a combination of environmental adaptation, natural selection, and genetic mutation. These processes allowed humans to develop and thrive in the African continent, and allowed for new traits and adaptations to develop over time.
Agreement Candidate 2	All three opinions agree that humans evolved through a combination of environmental conditions, natural selection, and genetic mutation. These processes are all intertwined and have helped shape the modern human form

and 2) an adapter module. During the fine-tuning process, only the adapter module's parameters are updated, while the original model's parameters remain unchanged. Both components contribute to the inference process.

Each instance in the training dataset consists of an "Instruction", an "Input" and an "Output", as shown in Fig. 1. Focusing on the task of consensus building, we set the instruction as "Find an agreement among the following opinions". This fine-tuning process equips the language model with the capability to efficiently identify consensus among diverse opinions.

4 Evaluation

4.1 Evaluation Setting

Dataset. In this paper, we use Yahoo! Answers topic classification dataset [11] which includes 1,400,000 training samples and 60,000 testing samples. Each sample includes following 5 parts: id, topic (class label) question title, question content and best answer. We topically choose the first 1000 question titles from training samples for generating training dataset. We then use each question content to generate opinions by using GPT-3. We consider both of the opinions have conflict and not, where the corresponding prompts are as follows, prompt1: *Generate* $|OP^i|$ *opinions for the topic of* $topic_i$ *which do not have a conflict* and prompt2: *Generate* $|OP^i|$ *opinions for the topic* of $topic_i$ which have a conflict. Then we use GPT-3 to generate an agreement by inputing the prompt as

Table 2. Four cases to fine-tune LLMs.

	Random Agreement Candidate	Optimal Agreement Candidate
Without conflict opinions	**Prompt**: Generate three opinions for the topic which do not have a conflict **Output**: Randomly choosing one from all agreement candidates as Output	**Prompt**: Generate three opinions for the topic which do not have a conflict **Output**: Optimally choosing one from all agreement candidates as Output
With conflict opinions	**Prompt**: Generate three opinions for topic which have a conflict **Output**:Randomly choosing one from all agreement candidates as Output.	**Prompt**: Generate three opinions for topic which have a conflict **Output**: Optimally choosing the one with maximized consensus score from all agreement candidates as Output

Please generate an agreement of the following opinions. independently for $|AC^i|$ times. Thus, for each question, we have $|OP^i|$ opinions and $|AC^i|$ agreement candidates. We set $|Q| = 1000$, $|OP^i| = 3$ and $AC^i = 4$ in this paper, which totally corresponds to 1000 questions, 6000 opinions (with conflict:3000, without conflict:3000) and 8000 agreement candidates.

Fine-tuning LLM. For fine-tuning the LLM, we select a 7 billion model from LLaMA [9] as a pre-trained model to be fine-tuned. We utilized the above dataset containing opinions with and without conflicts. Additionally, we divided it into two datasets based on the method of choosing an agreement candidate. We considered two approaches: selecting an optimal agreement candidate or randomly picking one for training. Consequently, this generated four distinct cases, as shown in Table 2.

Correspondingly, we employ GPT-3 as a baseline for comparison. For the test set, we randomly select 100 questions from the same dataset, excluding those used in the training dataset, to generate both opinions with conflicts and without conflicts.

4.2 Evaluation Results

We compare our methods with GPT-3 and present the results for the four cases depicted in Fig. 2. To calculate the average agreement score, agreements are generated from the 100 test samples, and each agreement score is calculated using the summation component in Eq. (1). First, when randomly selecting an agreement candidate as the final opinion summarization, Agreement-LoRA-7B (Self-Agreement) achieves comparable results in both cases of opinions with conflicts and without conflicts. Also, both Agreement-LoRA-7B (Self-Agreement) and GPT-3 exhibit lower scores for opinions with conflict than for opinions without conflict. This can be attributed to the fact that it is generally more challenging for agreement candidates to accommodate conflicting opinions.

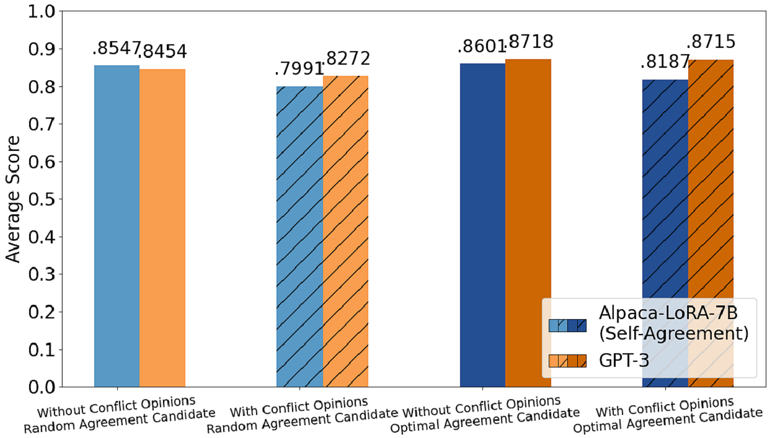


Fig. 2. The comparison of finding agreement results between GPT-3 and Agreement-LoRA-7B (Self-Agreement) in four cases. To calculate the average agreement score, agreements are generated from the 100 test samples, and each agreement score is calculated using the summation component in Eq. (1).

In the case where we select one optimal agreement candidate, the scores for both models improve. For instance, in the case of Agreement-LoRA-7B (Self-Agreement), the score increases to 0.86 from 0.85 in the situation without conflicting opinions and to 0.81 from 0.79 in the situation with conflicting opinions. These improvements highlight the benefits of selecting the optimal agreement candidate for opinion summarization tasks. Most notably, Agreement-LoRA-7B (Self-Agreement) demonstrates competitive performance in opinion summarization when compared to GPT-3, which has 175 billion parameters. This underscores the effectiveness of our Self-Agreement framework to fine-tune an LM model in handling diverse opinions.

Further, we list an instance of the above result. Table 3 shows the comparison results of the agreement candidates generated by GPT-3 models and Alpaca-LoRA (Self-Agreement), in response to a given topic: *Which is the country with top development indicators?*. The opinions represent three different views on which country has the top development indicators. In the GPT-3 generated agreement candidate, the model introduces India, China, and The United States as countries with high development indicators. Although these countries are well-regarded for their development, the response deviates from the given opinions. This indicates that GPT-3 may not effectively capture the essence of the conflicting opinions in this scenario. However, the Agreement-LoRA-7B (Self-Agreement) generated agreement candidate acknowledges all three opinions and successfully incorporates the key points from each. It emphasizes the strengths of India, China, and the United States while also recognizing that it is difficult to determine which country has the top development indicators, as each has its own strengths and weaknesses (shown in the highlighted parts of Table 3). This response demonstrates the effectiveness of the Agreement-LoRA-7B model framework provides a more balanced perspective by acknowledging the underly-

Table 3. One instance of comparing Agreement candidates of GPT-3 and Agreement-LoRA-7B (Self-Agreement) from opinions with conflict.

Topic	Which is the country with top development indicators?
Opinion 1	India is the country with top development indicators because it has achieved great progress in the fields of education, health, and poverty alleviation.
Opinion 2	China is the country with top development indicators because it has made great strides in economic growth and infrastructure development.
Opinion 3	The United States is the country with top development indicators because it has the highest GDP per capita and the most advanced technology.
Agreement Candidate (GPT-3)	We agree that India, China, and the United States are all countries with top development indicators. India has achieved great progress in the fields of education, health, and poverty alleviation; China has made great strides in economic growth and infrastructure development; and the United States has the highest GDP per capita and the most advanced technology. Together, these three countries demonstrate the highest level of development in the world.
Agreement Candidate (Self-Agreement)	We agree that India, China, and the United States are all countries with top development indicators. India has achieved great progress in the fields of education, health, and poverty alleviation, China has made great strides in economic growth and infrastructure development, and the United States has the highest GDP per capita and the most advanced technology. ***However, we disagree on which country has the top development indicators, as each country has its own strengths and weaknesses.***

ing disagreement and avoiding a definitive conclusion. This illustrates the effectiveness of the Self-Agreement framework in handling conflicting opinions and generating a consensus statement that fairly represents the diversity of viewpoints.

4.3 Fine-Tuning Analysis

The Agreement-LoRA-7B model is fine-tuned based on the pre-trained LLaMA-7B model. Our analysis focuses on how this fine-tuning under the Self-Agreement framework has impacted the performance, comparing it with the pre-trained LLaMA-7B model. Specifically, we examine the effects on domain-specific tasks and general tasks.

For domain-specific tasks, we assess the extent of performance improvement for generating agreement tasks. For general tasks, we consider the model's performance after fine-tuning. A critical concern is catastrophic forgetting, a phenomenon where a model, after being trained on a new task, significantly degrades its generalization performance on the original task, leading to a substantial loss of previously learned knowledge.

4.3.1 Domain-Specific Task Comparison

We utilize the data from the test dataset, with instructions and input (opinions) serving as prompts for testing. We discovered that the pre-trained LLaMA-7B model could only replicate the opinions without generating a new opinion. Table 4 provides an example about the topic of "What's your favorite Christmas memory". This demonstrates the significant improvement brought by our Self-Agreement fine-tuning in finding agreement among diverse opinions.

Table 4. Compare domain-specific instruction responses between Agreement-LoRA-7B and LLaMA-7B

Topic	What's your favorite Christmas memory?
Opinion 1	My favorite Christmas memory was when my family gathered around the dinner table to have a big feast and exchange gifts.
Opinion 2	My favorite Christmas memory was when I stayed up all night playing video games with my friends.
Opinion 3	My favorite Christmas memory was when I went to church with my family to celebrate the birth of Jesus.
LLaMA-7B	1. My favorite Christmas memory was when my family gathered around the dinner table to have a big feast and exchange gifts. 2. My favorite Christmas memory was when I stayed up all night playing video games with my friends. 3. My favorite Christmas memory was when I went to church with my family to celebrate the birth of Jesus.
Self-agreement	We agree that our favorite Christmas memories include gathering around the dinner table to have a big feast and exchange gifts, staying up all night playing video games with our friends, and going to church with our families to celebrate the birth of Jesus

4.3.2 General Task Comparison

To assess performance for general tasks and detect any signs of catastrophic forgetting, we use instruction data from the Self-Agreement framework. We find that the Agreement-LoRA-7B model, even after fine-tuning with the Self-Agreement framework, still exhibits a solid response to general instructions. Some instances are listed in Table 5. We further note that the Agreement-LoRA-7B model can provide more detailed responses than the LLaMA-7B model, even though such instructions do not directly feature in the dataset. For example, when prompted with "What are the three primary colors?", both models correctly identify the three primary colors, but our Agreement-LoRA-7B model supplements this by explaining, "These colors can be used to create any other color by mixing them together". Furthermore, for the 4th prompt "Instruction: Identify the odd one out. Input: Twitter, Instagram, Telegram", while the LLaMA-7B model gives an incorrect answer, the Agreement-LoRA-7B model correctly identifies the outlier and further elaborates on its choice.

Table 5. Compare general instruction responses between Agreement-LoRA-7B and LLaMA-7B.

Prompt	Completion(LLaMA-7B)	Completion(Agreement-LoRA-7B)
Instruction: Give three tips for staying healthy.	\begin{itemize} \item Eat a balanced diet. \item Exercise regularly. \item Get enough sleep. \end{itemize}	1. Eat a balanced diet of fruits, vegetables, and whole grains. 2. Exercise regularly to keep your body in shape. 3. Get enough sleep to recharge your body and mind.
Instruction: What are the three primary colors?	The three primary colors are red, blue, and yellow.	The three primary colors are red, blue, and yellow. These colors can be used to create any other color by mixing them together.
Instruction: Describe the structure of an atom.	An atom is made up of a nucleus, electrons, and neutrons. The nucleus is made up of protons and neutrons. Protons have a positive charge and neutrons have no charge. Electrons have a negative charge	An atom is made up of a nucleus surrounded by a cloud of electrons. The nucleus is made up of protons and neutrons, while the cloud of electrons is made up of negatively charged electrons.
Instruction: Identify the odd one out. **Input** Twitter, Instagram, Telegram	Twitter	Telegram is the odd one out as it is a messaging app, while Twitter and Instagram are social media platforms

5 Conclusion

Developing LLM for consensus-building usually comprises two main phases: training and deployment. During training, we aim to refine our LLM to produce agreement candidates likely to be accepted by humans. The deployment phase tests the LLM's capability in real human discussions. While we recognize the value of human feedback, our focus in this paper is on the training phase. This is to avoid potential biases that might limit the model's broad applicability.

Specifically, we introduced the Self-Agreement, an efficient framework to fine-tune LLMs to autonomously find agreement among diverse opinions. Our method eliminates the need for expensive human-generated data. We also presented a large dataset of questions, opinions, and agreement candidates, serving as a valuable resource for future consensus-building models. Since it is hard to identify the preference for each users, we use the similarity calculated by the BERT-based model. Although such kinds of similarity can somehow reflect how the agreement candidate match opinion, there is a gap which should be tackled as our future work. Our experiments highlight the effectiveness of our framework in consensus-building tasks while achieving comparable performance to GPT-3 with only 1/25 of its parameters.

Acknowledgement. This work was supported by a CREST Grant (JPMJCR20D1) from Japan Science and Technology Agency (JST) and a Grant-in-Aid for Scientific Research (C) (23K11230) from the Japan Society for the Promotion of Science (JSPS).

References

1. Bakker, M., et al.: Fine-tuning language models to find agreement among humans with diverse preferences. In: Advances in Neural Information Processing Systems, vol. 35, pp. 38176–38189 (2022)
2. Broom, M.: A further study of the validity of a test of social intelligence. J. Educ. Res. **22**(5), 403–408 (1930)
3. Brown, T., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
4. Conzelmann, K., Weis, S., Süß, H.M.: New findings about social intelligence. J. Individ. Differ. (2013)
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
6. Reinald Kim Amplayo, M.L.: Informative and controllable opinion summarization (2021). https://arxiv.org/pdf/1909.02322.pdf
7. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
8. Suhara, Y., Wang, X., Angelidis, S., Tan, W.C.: OpinionDigest: a simple framework for opinion summarization. arXiv preprint arXiv:2005.01901 (2020)
9. Touvron, H., et al.: LLaMA: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
10. Wang, Y., et al.: Self-instruct: aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560 (2022)
11. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, vol. 28 (2015)

Self-SLP: Community Detection Algorithm in Dynamic Networks Based on Self-paced and Spreading Label Propagation

Zijing Fan[1,2](✉) and Xiangyu Du[1,2,3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{fanzijing, duxiangyu}@iie.ac.cn
[3] Beijing Trusfort Technology Co. Ltd., Beijing, China

Abstract. With the continuous expansion of network size, existing complex networks have dynamic characteristics gradually. The effective detection of communities in dynamic networks has become a current research hotspot. Detection methods based on label propagation are relatively mature and classical. However, they failed to address the instability issue caused by the randomness of propagation itself. Furthermore, the state-of-the-art methods ignore the learning ability of the algorithm itself and do not have a validation module. Therefore, we propose a self-paced and spreading label propagation algorithm (Self-SLP) for community detection in dynamic networks. To prevent the consumption of computational resources due to random propagation, we design a self-paced spreading activation algorithm. On this basis, we propose belonging coefficient difference for validation, which improves the stability and reliability of our algorithm. To the best of our knowledge, we are the first to consider this idea of self-learning to improve community detection. In contrast, the method proposed in this paper makes propagation more flexible while limiting excessive randomness. Experimental results on large-scale real-world and synthetic networks show that Self-SLP performs well for community detection in dynamic networks and confirms computational efficiency and reliability.

Keywords: Community detection · Dynamic network · Spreading Label Propagation · Self-paced learning · Belonging coefficient difference

1 Introduction

In complex network analysis, community detection is a crucial task. The community structure is generally understood as a collection of closely connected nodes in the network, where internal edges are dense and external edges are sparse. Community detection involves employing a range of methods to identify closely

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 310–322, 2024.
https://doi.org/10.1007/978-981-99-7022-3_27

connected nodes. Its applications are diverse; for instance, social networks can employ community detection to identify social circles, while protein networks can identify specific protein modules [1].

Static network community detection needs to consider the characteristics of the network data for supervised or unsupervised clustering of closely linked nodes, and common methods include modularity-based methods [2], network embedding [3], and so on. And dynamic network association detection also needs to consider the node-link tightness of each time snapshot.

Label Propagation (LP) is an efficient method to detect community, which updates the label of each node by replacing it with the label used by the node with the largest number of neighbors. The improved method based on LP is suitable for discovering communities in dynamic networks [4]. In addition, the community results are highly randomized. Especially when updating asynchronously, changes in the update order can also lead to different community detection results. Aiming at this problem, [5] proposed a spreading activation label propagation method, which assigns an activation value to each node and propagates through a spreading activation process and LP.

Despite these successes, the research on community detection has never taken into consideration the concept of self-learning, which could potentially limit the random propagation of the label propagation. This prompts us to investigate the learning capabilities of the propagation algorithm. Our newly introduced Self-Paced and Spreading Label Propagation Algorithm (Self-SLP) utilises three innovative strategies: the self-paced approach, the spreading label propagation method and belonging verification strategy.

Specifically, the main contributions of our study are as follows:

- We design a soft spreading strategy for propagation, which reduces the label oscillation caused by the inherent random strategy of label propagation.
- We introduce a self-paced learning with diversity algorithm to alleviate the randomness in the label propagation process without reducing its efficacy.
- We design a validation module to enhance the reliability of the community detection algorithm through proposing belonging verification difference.

This paper is arranged as follows. In Sect. 2, we review the definitions of the proposed algorithm and related work about community detection. Then, we expatiate the details of our proposed Self-SLP in Sect. 3. Section 4 reports the experimental results and the conclusion is drawn in Sect. 5.

2 Background and Related Work

In this section, we introduce the background information of community detection in dynamic networks and self-paced learning with diversity. Then we review related prior research.

2.1 Community Detection in Dynamic Network

Let $G = (V, E)$ be a graph, with $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_m\}$, $n = |V|$, $m = |E|$. The adjacency matrix of G is denoted by $A = (A_{ij})$, where $A_{ij} = 1$ if there is an edge between node i and node j, and $A_{ij} = 0$ otherwise.

The dynamic complex network adds the attribute of time t, which can be abstracted by a series of network snapshots $\mathbb{G} = \{G^{(0)}, G^{(1)}, ..., G^{(S)}\}$, where $G^{(t)} = (V^{(t)}, E^{(t)})$ is the snapshot point of the network at that time $0 \leq t \leq s$. The change between two consecutive snapshots $G^{(t)}$ and $G^{(t-1)}$ is denoted by $\Delta G^{(t)} = (\Delta V^{(t)}, \Delta E^{(t)})$ where $\Delta V^{(t)} = V^{(t)} \ominus V^{(t-1)}$ and $E^{(t)} = E^{(t)} \ominus E^{(t-1)}$.

2.2 Self-paced Learning with Diversity

Self-paced learning (SPL) [6] is a learning mechanism, in which complex examples are gradually incorporated into the training. It facilitates data-efficient learning [7], adversarial robustness [8] and positive-unlabeled learning [9]. Self-paced learning with diversity (SPLD) [10] solves the problem of sample selection diversity in self-paced learning. Here, diversity refers to the diversity of samples selected by self-paced. Diversity tends to select samples with low similarity and large diversity between samples, which is suitable for multi-classification tasks such as community detection.

Suppose that the sample set $X = (x_1, x_2, ..., x_n) \in R_{m\times n}$ can be divided into b clusters. That is $X^{(1)}, ..., X^{(b)}$, Where $X^{(j)} \in R_{m\times n_j}$ represents the sample set belonging to the j-th cluster and n_j represents the number of samples of the j-th cluster. Suppose that the model is $f(x)$, and the loss function is $L(f(x), y)$, where $f(x)$ is the predicted output value of the corresponding sample x, and y is the true label value of the sample x. The objective function of self-paced learning with diversity can be defined as follows:

$$\underset{\mathrm{w,v}}{min}\mathbb{E}(\mathrm{w}, \mathrm{v}; \lambda, \gamma) = \sum_{i=1}^{n} v_i L(y_i, f(\mathrm{x_i}, \mathrm{w})) - \lambda \sum_{i=1}^{n} v_i - \gamma ||\mathrm{v}||_{2,1} \tag{1}$$

where $\mathrm{v} \in [0, 1]^n$, λ and γ represent easy sample items and diverse sample items, respectively.

2.3 Prior Research

SLPD [11], as a dynamic version of the SLP method, randomly selected receiver node and its neighbor nodes as the speaker to spread labels. K. Liu et al. proposed DLPE [12], which decides the label through the neighbors of the node and attaches confidence to each neighbor. H. Zhang et al. also proposed DSLPA [13], which improved the SLPA algorithm by using the history label. AC2CD [14] by A. Costa et al. uses a deep reinforcement learning strategy for regional optimisation of modular density functions to identify dynamic associations. SALP [5] addressed the problem of LP creating unwieldy communities in dynamic social networks by assigning an activation value to nodes while constructing two weight variants.

3 The Proposed Method

The method proposed in this paper fully utilises the learning ability of label propagation, which has three components. Firstly, the soft spreading strategy is introduced. Next, a self-paced learning algorithm with diversity is designed to reduce the randomness in the label propagation process. In the third section, we propose using the belonging coefficient difference to determine the nodes at the boundary between communities, improving the algorithm's stability and reliability. The Self-SLP framework's architecture is depicted in Fig. 1.

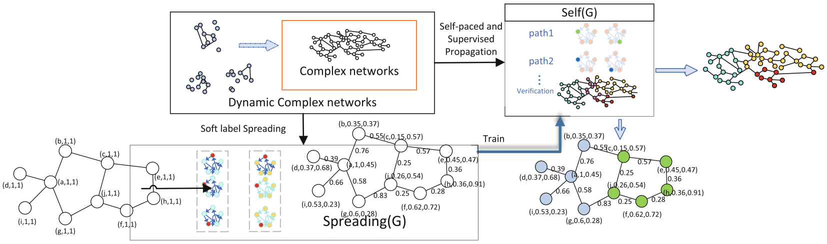


Fig. 1. Overview of the community detection framework using self-paced and label propagation (Self-SLP).

3.1 Soft Spreading Strategy

To minimize the impact of the avalanche effect induced by the random strategy of label propagation, we propose utilizing soft spreading in propagation. Specifically, we commence by assigning label matrices (label name, soft label value, activation value) to designated nodes. To decrease resource consumption during propagation, we opt to choose nodes with degrees in the top 10% rather than every node, as in the study by [5], which utilises a scale-free architecture where nodes have a degree of 4. The selected source nodes are labelled as S, with a label matrix of $(1, 1, 1)$ and the remaining nodes as $(0, 0, 1)$. Each node modifies or includes its label matrix based on neighbour information. The recipient node updates its label matrix in accordance with Eq.(2) and (3).

$$S[j] = \sum_{i \in n(j)} (S[i] * P[i,j] * D_1) \tag{2}$$

$$A[j] = A[i] + \sum_{i \in n(j)} (A[i] * W[i,j] * D_2) \tag{3}$$

where $S[o]$ and $A[o]$ represent the soft label and activation values of nodes i. Symbol $D_i, i \in \{1, 2\}$ is the decay factor, and $P[i, j]$ and $W[i, j]$ represent the transition probability and weight between nodes i and j. When using Algorithm 1, we set α as the parameter for depth-first traversal.

Algorithm 1: Soft-Spreading Algorithm: Soft-Spreading(G)

```
Input: $G = (V, E), \alpha, D_1, D_2$
Output: $G = (V', E)$
begin
    for $v \in V$ do
        Create set $\Lambda = \{\}$
        Add $v$ and $v.neighbours$ to $\Lambda$
        for $i = 0 \rightarrow \alpha$ do
            Select $nv$ from $\Lambda$
            Computer $nv.softlabel$ based on Eq.2
            Computer $nv.label.activationvalue$ based on Eq.3
            if $nv.label.activationvalue \geq threshold$ then
                Add $nv.Neighbours$ to $\Lambda$
return $G = (V', E)$
```

3.2 Self-paced Propagation Learning with Diversity

Self-paced learning with diversity (SPLD) seeks to learn varying information from data and compensate for the absence of supervision when obtaining precise annotations of large amounts of unlabeled data proves challenging. In view of this, to address the instability of the LP-based method created by the randomness of propagation without compromising its effectiveness, we adopt self-paced propagation learning with diversity to progressively partition communities. To the best of our knowledge, this is the first time that self-learning techniques have been implemented to enhance community detection. The algorithmic process is outlined in Algorithm 2.

Algorithm 2: SPLDWeighting(G)

```
Input: Dataset $\mathcal{D}, groupsX^1, ..., X^b$, w; $\lambda, \gamma$
Output: Global solution $v = (v^1, ..., v^b)$ of $min_v\mathbb{E}(\mathrm{w}, \mathrm{v}; \lambda, \gamma)$
begin
    for $j = 1$ to $b$ do
        Sort the samples in $X^{(j)}$ as $(X_1^{(j)}, ..., X_{n_j}^{(j)})$ in ascending order of
        their loss values $L$
        Accordingly, denote the labels and weights of $X^{(j)}$ as $(y_1^{(j)}, ..., y_{n_j}^{(j)})$
        and $(v_1^{(j)}, ..., v_{n_j}^{(j)})$; for $i = 1$ to $n_j$ do do
            if $L(y_i^{(j)}, f(x_i^{(j)}, w)) < \lambda + \gamma\frac{1}{\sqrt{i}+\sqrt{i-1}}$ then
                $v_i^{(j)} = 1$
            else
                $v_i^{(j)} = 0$
    return $v$
```

3.3 Belonging Verification

Considering the instability of community detection due to the growth of dynamic networks, we introduce verification to adjust the division. Taking into account the changes in dynamic community nodes, [15] proposed the use of a belonging coefficient to measure the link strength between nodes. Building on this, we propose using the belonging coefficient difference to verify the nodes in community boundaries. This improves the stability and reliability of the algorithm.

$$BC_d(v) = \sum_{x_i, x_j \in C^*} || \frac{\sum_{x_i \in N_i'} \frac{k_{int}^{x_i}}{|N(x_i)|} - \sum_{x_j \in N_j'} \frac{k_{int}^{x_j}}{|N(x_j)|}}{|N(v)|} || \quad (4)$$

where $N_i^{'} = N(x_i) \cap C, N_j^{'} = N(x_j) \cap C$, and C^* represents the detected communities.

In brief, to mitigate label propagation's randomness and enhance its dependability, we propose using a soft spreading strategy and adopting self-paced propagation learning with diversity. This constrains the propagation process by minimizing the loss function. Additionally, we optimize the algorithm parameters by comparing the belonging coefficient differences. The complete algorithm is presented in Algorithm 3.

Algorithm 3: Self-SLP

```
Input: $\{G_1 =< V_1, E_1 >, G_2 =< V_2, E_2 >, \cdots, G_T =< V_T, E_T >\}, T, \beta$
Output: Set of communities of $G_n$
begin
    for $ts = 1 : T$ do
        $\mathcal{G}, \alpha$=SPLDWeighting(Soft-Spreading(G));
        for $i, j = \{1, 2, \cdots, T\}$ do
            if $G_i =< V_i, E_i > \in \mathcal{G}$ then
                SPLDWeighting(G);
                if $|V_T| > |V_{T-1}|$ or $|E_T| > |E_{T-1}|$ then
                    Self-SLP(G,T);
                if $BC_d(v) < \beta$ then
                    return;
    return $\{G_1, G_2, \cdots, G_c\}$
```

3.4 Time Complexity

In the soft spreading strategy, it takes $O(n)$ to initialize the labels, and at most $O(2m)$ for soft labelling and spreading. It requires a total of $O(n * m)$. In the belonging verification part, we need to calculate the difference of the belonging coefficients which takes $O(k * n_C)$, where k is the average degree of G. In the self-paced diversity phase, the time complexity is $O(b * n_j)$. Moreover, in the snapshot increase phase, $O(T*T)$ is required. Therefore, the total time complexity of Self-SLP is $O(n*m)$.

4 Experiment

In the experimental section, we assess the effectiveness of Self-SLP on four actual networks and four synthetic networks. Our algorithms have been implemented using Python 3.8 with NetworkX, sci-kit learn, numpy and matplotlib libraries. The evaluation of our algorithms has been conducted on a server with 128GB of memory and two Intel Xeon CPUs E5-2640. To ensure accuracy and stability of the experimental results, each experiment has been run ten times, and the resulting average values have been taken.

4.1 Baseline Methods

We select SLPD [11], DLPE [12], DSLPA [13], AC2CD [14], RWSALP [5] and SBSALP [5] as the baseline methods for the Self-SLP algorithm since they can detect communities in dynamic networks.

SLPD: It is a dynamic community discovery algorithm based on the speaker-listener label Propagation.

DLPE: It is an evolutionary clustering method that assigns a community label to a node based on its neighbors and also assigns a confidence coefficient to each neighbor.

DSLPA: It is a multi-label propagation algorithm that uses the history label to initialize the labels in community discovery.

AC2CD: The approach utilises local optimisation of modular density functions to dynamically detect associations under a deep reinforcement learning strategy.

RWSALP: It uses a weighting method, known as a random walk, which assigns an activation value to each of the labels.

SBSALP: It utilises a social behavioural weighting technique that assigns an activation value to each label.

4.2 Quality Measurement

Modularity [16] quantifies the quality of network partition compared to the entire network. A higher modularity indicates a superior division of the network.

$$Q = \frac{1}{4m} \sum_{ij} (A_{i,j} - \frac{k_i k_j}{2m}) s_i s_j \tag{5}$$

Where A_{ij} represents the number of edges between node i and node j in the adjacency matrix A. For binary classification, assign s_i the value of 1 if vertex i belongs to group 1 and -1 if it belongs to group 2.

The Normalized Mutual Information (NMI) is a potent index adopted in various community discovery algorithms to appraise the likeness of identified communities and actual communities. The NMI value is directly proportional to the relatedness of both communities. When there are actual communities present in the network, NMI proves to be the most accurate and resilient evaluation metric.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{|A|}\sum_{j=1}^{|B|} N_{ij} log(\frac{N_{ij}N}{N_{i_s}N_{j_s}})}{\sum_{i=1}^{|A|} N_{i_s} log(\frac{N_{i_s}}{N}) + \sum_{j=1}^{|B|} N_{j_s} log(\frac{N_{j_s}}{N})} \tag{6}$$

Where $A = \{a_1, a_2, ...a_k\}$ represents the standard division labelled by real-world communities and $B = \{b_1, b_2, ...b_l\}$ denotes the detected division obtained by a community detection algorithm. N is the confusion matrix with rows and columns corresponding to the "real world" and "detected" communities, respectively.

Table 1. Introduction to the Real-world Networks Dataset

Dataset	Type	Content
Arxiv HEP-PH	Citation networks	Articles published between January 1993 and April 2003
Arxiv HEP-TH	Citation networks	High-energy physics theory publications from January 1993 to April 2003
Enron Email	Email network	Enron Email dataset over 15 years
DBLP	Bibliography network	Computer science bibliography providing a comprehensive list of research papers in computer science

4.3 Experiments on Real-World Networks

We carried out algorithm evaluation experiments on four real-world networks: Arxiv HEP-PH, Arxiv HEP-TH, Enron Email, and DBLP. The datasets are described in detail in Table 1 and Table 2. Parameters were set at $\alpha = 3, D_1 = 0.55, D_2 = 0.41, \beta = 0.2$, following iterative result training.

Experimental Results and Analysis: The comparison of modularity between the proposed method Self-SLP and other available methods is presented in Table 3. The results derived from the Arxiv HEP-PH dataset indicate a declining trend in the modularity of all methods with the increasing number of snapshots. It is normal and unavoidable for the algorithm to account for changes in both edges and nodes as part of the process. Furthermore, the decline rate of modularity in Self-SLP gradually slows down, resulting in a higher overall modularity value compared to other methods. This is attributed to the algorithm's ability to adapt as the number of early snapshots increases, leading to increased performance robustness. Furthermore, the proposed method demonstrates more

Table 2. Overview of the Real-world Networks Dataset

Dataset	Number of nodes	Number of edges	Number of snapshots	Snapshots frequency
Arxiv HEP-PH	34546	421578	124	one month
Arxiv HEP-TH	27770	352807	62	two months
Enron Email	36692	367662	90	two months
DBLP	317080	1049866	365	ten days

stable detection performance compared to other methods that show clear points of inflection in modularity.

On the Arxiv HEP-TH dataset, Self-SLP shows high modularity in most snapshots, especially in later ones. Additionally, when there are many snapshots, Self-SLP's modularity decreases more slowly and is clearly superior to that of other methods. Moreover, as the number of snapshots increases, the overall modularity decreases at the slowest rate.

Table 3. Modularity values of Self-SLP, RWSALP, SBSALP, AC2CD, DSLPA, DLPE and SLPD on Arxiv HEP-PH, Arxiv HEP-TH, Enron and DBLP.

Dataset	Snapshot	Self-SLP	RWSALP	SBSALP	AC2CD	DSLPA	DLPE	SLPD
Arxiv HEP-PH	30	**0.8623**	0.6883	0.6764	0.6883	0.6861	0.6799	0.6962
	60	**0.8451**	0.6693	0.6664	0.6729	0.6652	0.6499	0.6462
	90	**0.8346**	0.6791	0.6789	0.6632	0.6447	0.6189	0.6169
	120	**0.8232**	0.6483	0.6264	0.6401	0.6399	0.5529	0.5862
Arxiv HEP-TH	15	**0.8451**	0.6983	0.6465	0.6819	0.6872	0.6601	0.6462
	30	**0.8218**	0.6734	0.6328	0.6698	0.6628	0.5779	0.5852
	45	**0.8171**	0.6493	0.6169	0.6421	0.6595	0.5731	0.5369
	60	**0.8159**	0.6281	0.6164	0.6313	0.6477	0.5599	0.5032
Enron	20	**0.8551**	0.6783	0.6463	0.6443	0.6329	0.6127	0.6062
	40	**0.8359**	0.6681	0.6204	0.6392	0.6296	0.5759	0.5757
	60	**0.8236**	0.6553	0.6164	0.6296	0.6037	0.5751	0.5561
	80	**0.8212**	0.6189	0.6153	0.6011	0.5961	0.5669	0.5692
DBLP	50	**0.8458**	0.7883	0.7664	0.7688	0.7791	0.7292	0.7362
	100	**0.8371**	0.7599	0.7689	0.7541	0.7203	0.6761	0.6771
	150	**0.8299**	0.7511	0.7469	0.7513	0.7059	0.6193	0.6699
	200	**0.8151**	0.7485	0.7411	0.7443	0.6891	0.6099	0.6168

On the Enron email network, it is clear that Self-SLP achieves greater modularity than other methods in all snapshots. This indicates that the proposed Self-SLP method uncovers more accurate and consistent communities than other methods for larger communities. Furthermore, addition or removal of nodes from the Enron email data does not have a negative effect on the proposed method's performance.

As a consequence of the experiment conducted on DBLP, the performance of Self-SLP aligns with that of RWSALP and SBSALP when the snapshot time is brief on the dataset containing a considerable number of nodes and snapshots. As time passes, our suggested method gradually surpasses other techniques, especially SLPD and DLPE.

Table 4. Average running time(sec) achieved by SLPD, DLPE, DSLPA, AC2CD, SBSALP, RWSALP and Self-SLP on real networks.

Datasets	SLPD	DLPE	DSLPA	AC2CD	SBSALP	RWSALP	Self-SLP
Arxiv HEP-PH	176	211	531	1486	6291	6438	871
Arxiv HEP-TH	149	176	472	1169	5709	5997	732
Enron Email	262	218	386	1531	7900	8424	513
DBLP	569	598	621	1744	35498	36712	899

In Table 4, the average duration of Self-SLP and other available methods on real networks is displayed. The figures in Table 4 suggest that the suggested approach's community detection is slower than SLPD and DLPE, yet markedly quicker than SBSALP and RWSALP. Our technique leverages self-learning, offering adaptability to dynamism in real-world community networks, ultimately resulting in heightened robustness and consistency.

4.4 Experiments on Synthetic Networks

The LFR network [17] exhibits significant characteristics of real-world complex networks. Its node degree and association size distribution adhere to a scale-free pattern, characterised by an adjustable exponential power law. This network model is suited to evaluate diverse networks. This investigation evaluates the proposed algorithm employing four sets of synthetic networks, each with real-world community structures. In Table 5, the primary parameters utilised for the synthetic networks are presented. To specify: α is set to 4, D_1 is set to 0.52 and D_2 is set to 0.48. Meanwhile, β is set to 0.3 by way of iterative result training.

Experimental Results and Analysis: As shown in Table 6: (a) In the case of Birth/Death, our method exhibits a considerable performance advantage over DLPE and SLPD when the snapshot time is small. While the time increases, Self-SLP maintains a stable performance and supersedes SLPD by a large margin. (b) For Birth/Expand, Self-SLP outperforms all other methods. As time progresses, the performance of Self-SLP decreases more evenly. (c) In the case of Expand/Contract, the impact of Self-SLP is particularly pronounced, and its stability is high, especially as the snapshot size increases. This is evident in the fact that the proposed method presents a significant improvement in comparison

Table 5. Parameters of LFR benchmark networks

Parameters of LFR	Explanation	Value
n	Number of nodes	vary
on	Number of nodes	vary
om	Number of memberships	vary
mu	Mixing parameter	vary
d_{avg}	Mixing parameter	20
d_{max}	Maximum degree	50
t_1	Exponent for node degree distribution	2
t_2	Exponent for community size distribution	1
C_{min}	Minimum community size	$n/50$
C_{max}	Maximum community size	$n/10$

Table 6. NMI values of Self-SLP ,RWSALP, SBSALP, AC2CD, DSLPA, DLPE and SLPD on Synthetic networks. (a) Birth/Death (b) Birth/Expand (c) Expand/Contract (d) Expand/Death.

Dataset	Snap-shot	Self-SLP	RWSALP	SBSALP	AC2CD	DSLPA	DLPE	SLPD
Birth/Death	2	**0.7751**	0.6383	0.6664	0.6533	0.6538	0.6149	0.5662
	4	**0.7439**	0.6181	0.6359	0.6317	0.6217	0.5951	0.5561
	6	**0.7128**	0.5853	0.5681	0.6086	0.6088	0.5438	0.5371
	8	**0.7057**	0.5807	0.5474	0.6086	0.5726	0.5279	0.5169
Birth/Expand	2	**0.7636**	0.6350	0.6064	0.6199	0.6136	0.5848	0.5862
	4	**0.7201**	0.6019	0.5709	0.5826	0.5957	0.5355	0.5348
	6	**0.7192**	0.6008	0.5638	0.5673	0.5695	0.5351	0.5091
	8	**0.6838**	0.5781	0.5489	0.5673	0.5261	0.5122	0.5355
Expand/Contract	2	**0.7651**	0.6155	0.6355	0.6399	0.6238	0.6009	0.5804
	4	**0.7488**	0.6386	0.6055	0.6018	0.6096	0.5782	0.5521
	6	**0.7155**	0.5744	0.5659	0.5611	0.5437	0.5458	0.5499
	8	**0.6452**	0.5159	0.5118	0.5304	0.5273	0.5137	0.5368
Expand/Death	2	**0.7779**	0.6499	0.6368	0.6427	0.6248	0.6099	0.5862
	4	**0.7443**	0.6383	0.6093	0.6003	0.5931	0.5791	0.5638
	6	**0.7199**	0.5559	0.5429	0.5836	0.5782	0.5675	0.5218
	8	**0.6887**	0.5506	0.5351	0.5406	0.5633	0.5392	0.5432

to other methods, exhibiting minimal NMI fluctuations, particularly in scenarios where other methods experience multiple turning points.

Based on our analysis, it can be concluded that Self-SLP outperforms other methods in dynamic community detection tasks on both real-world and synthetic networks. The experimental findings indicate an increase in modularity of the method proposed by 26.5% on average while the NMI sees an average increase of 24.6%. Additionally, Self-SLP algorithm demonstrates efficient rates in dynamic community detection.

5 Conclusion

To solve the problem of community detection in the dynamic network, this paper proposes a Self-SLP algorithm. In Self-SLP, a label propagation technique that utilises self-paced and spreading is proposed as a solution to the instability issue triggered by the arbitrary propagation of LP. The experimental study demonstrates the effectiveness and efficiency of the Self-SLP algorithm in both dynamic real-world and synthetic networks. Moving forward, we will continue to investigate this technique to enhance the efficiency of valuable community detection in more intricate and variable networks, possibly by combining it with local dynamic extension and reduction methods.

Acknowledgment. We would like to thank the anonymous reviewers for their insightful comments. This research is supported by National Natural Science Foundation of China (No. 62202466), Youth Innovation Promotion Association CAS (No. 2022159), and the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02030200). This research is also supported by the Key Laboratory of Network Assessment Technology of Chinese Academy of Sciences, and the Beijing Key Laboratory of Cyber Security.

References

1. Ma, X., Zhang, B., Ma, C., Ma, Z.: Co-regularized nonnegative matrix factorization for evolving community detection in dynamic networks. Inf. Sci. **528**, 265–279 (2020)
2. Gsgens, M., van der Hofstad, R., Litvak, N.: The hyperspherical geometry of community detection: modularity as a distance. J. Mach. Learn. Res. **24**(112), 1–36 (2023)
3. Zhou, X., Su, L., Li, X., Zhao, Z., Li, C.: Community detection based on unsupervised attributed network embedding. Expert Syst. Appl. **213**, 118937 (2023)
4. Liu, K., Huang, J., Sun, H., Wan, M., Qi, Y., Li, H.: Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks. Knowl.-Based Syst. **89**, 487–496 (2015)
5. A spreading activation-based label propagation algorithm for overlapping community detection in dynamic social networks. Data & Knowledge Engineering, vol. 113, pp. 155–170 (2018)
6. Kumar, M., Packer, B., Koller, D.: Self-paced learning for latent variable models. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 23. Curran Associates Inc., (2010)
7. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 4037–4058 (2021)
8. Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., Wang, Z.: Adversarial robustness: From self-supervised pre-training to fine-tuning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020, pp. 696–705 (2020)
9. Chen, X., et al.: Self-PU: self boosted and calibrated positive-unlabeled training. In: Proceedings of the 37th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1510–1519

10. Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., Hauptmann, A.: Self-paced learning with diversity. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates Inc., (2014)
11. Aston, N., Hu, W.: Community detection in dynamic social networks. Commun. Network **6**(2), 124–136 (2014)
12. Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks. Knowledge-Based Systems, vol. 89, pp. 487–496 (2015)
13. Zhang, H., Dong, B., Wu, H., Feng, B.: A multi-label propagation community detection algorithm for dynamic complex networks. In: International Conference on Advanced Information Systems Engineering, pp. 467–482. Springer (2021)
14. Costa, A.R., Ralha, C.G.: Ac2cd: an actor-critic architecture for community detection in dynamic social networks. Knowl.-Based Syst. **261**, 110202 (2023)
15. Cheng, F., Wang, C., Zhang, X., Yang, Y.: A local-neighborhood information based overlapping community detection algorithm for large-scale complex networks. IEEE/ACM Trans. Networking **29**(2), 543–556 (2021)
16. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. U.S.A. **103**(23), 8577–8582 (2006)
17. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure of complex networks. New J. Phys. **11**(3) (2009)

Word Segmentation of Hiragana Sentences Using Hiragana BERT

Jun Izutsu[1], Kanako Komiya[2](✉), and Hiroyuki Shinnou[1]

[1] Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki 316-0033, Japan
{21nm707h,hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp
[2] Tokyo University of Agriculture and Technology, 2-24-16 Nakaco, Koganeishi, Tokyo 184-8588, Japan
kkomiya@go.tuat.ac.jp

Abstract. Unlike Western languages, word segmentation is necessary for Japanese sentences because they do not have word boundaries. The performances of existing morphological analyzers for Japanese sentences are very high. However, it is difficult to segment sentences mostly written in Hiragana, which is a Japanese writing system simpler than Kanji, because clues to segment the sentences decrease. In this study, we created a word segmentation model of Hiragana sentences using two types of BERT: unigram and bigram BERT models. We pre-trained the BERT models with Wikipedia and fine-tuned them with the core data of the Balanced Corpus of Contemporary Written Japanese for word segmentation. In addition to the two types of BERT-based word segmentation systems, we developed a word segmentation system for Hiragana sentences using KyTea, a toolkit developed for analyzing text, with a focus on languages requiring word segmentation. We compared them in word segmentation of Hiragana sentences. The experiments revealed that the unigram BERT-based word segmentation system outperformed the bigram BERT-based word segmentation system and the KyTea-based word segmentation system.

Keywords: Word segmentation · Japanese Hiragana · BERT

1 Introduction

Japanese sentences contain various kinds of characters, such as Kanji (Chinese character), Hiragana, Katakana, numbers, and alphabets, which makes it difficult to learn. Japanese speakers usually learn Hiragana first in their school days because the number of characters is much smaller than the Kanji; Hiragana has 46 characters, and Japanese uses thousands of Kanji. Most Japanese sentences are composed of all kinds of characters. These sentences are called Kanji-Kana mixed sentences. However, it is difficult for many non-Japanese speakers to learn thousands of Kanji, so children and new Japanese language learners use Hiragana sentences.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 323–335, 2024.
https://doi.org/10.1007/978-981-99-7022-3_28

Unlike Western languages, Japanese and Chinese do not have word boundaries, so word segmentation is necessary for the natural language processing of these languages. MeCab[1] and Chasen[2] are morphological analyzers for Japanese that segment Japanese sentences into words. The existing Japanese morphological analyzers' performances are very high. However, it is difficult to segment sentences written almost entirely in Hiragana[3] into words using these systems because they are designed for Kanji-Kana mixed sentences. When almost all sentences are written in Hiragana, which makes it challenging to identify the location of words to be segmented.

This paper developed two types of BERT [1] (Bidirectional Encoder Representations from Transformers) models for Hiragana sentences: unigram and bigram BERT models (see Sect. 3). We utilized a large amount of automatically tagged data for pre-training and used manually tagged data for fine-tuning (see Sect. 4). In addition, to compare with the performances of our Hiragana BERT-baed word segmentation models, we developed a Hiragana word segmentation model using KyTea[4], a toolkit developed for analyzing text, with a focus on languages requiring word segmentation. The experiments revealed that the unigram BERT model outperformed the bigram BERT model and KyTea model (see Sect. 6). We discussed the reason (see Sect. 7) and conclude this paper (see Sect. 8).

The contributions of this paper are as follows:

1. We developed Hiragana unigram and bigram BERT models,
2. We showed that automatically tagged data is effective for pre-training of Hiragana BERT models, and
3. We discussed why the Hiragana unigram BERT-based word segmentation model outperformed the Hiragana bigram BERT-based or KyTea-based word segmentation models.

2 Related Work

There are some studies on word segmentation and morphological analysis of Hiragana sentences, which include the following. First, Kudo et al. [5] modeled the process of generating Hiragana-mixed sentences, which are sentences that include words written in Hiragana but usually written in different characters such as Kanji, using a generative model. They proposed a method to improve the accuracy of parsing Hiragana-mixed sentences by estimating its parameters using a large Web corpus and an EM algorithm. Hayashi and Yamamura [2] reported that adding Hiragana words to the dictionary improves the accuracy of morphological analysis. Izutsu et al. [3] converted MeCab's ipadic dictionary into

[1] https://taku910.github.io/mecab/.
[2] https://chasen-legacy.osdn.jp.
[3] Numbers and symbols are included.
[4] http://www.phontron.com/kytea/.

Hiragana and used a corpus consisting only of Hiragana to perform morphological analysis of Hiragana-only sentences. In addition, Izutsu and Komiya [4] performed a morphological analysis of Hiragana sentences using the Bi-LSTM CRF model and reported how the accuracy of morphological analysis could be changed by training and fine-tuning over multiple domains of sentences. Moriyama et al. [6] also performed the morphological analysis of plain Hiragana sentences using the Recurrent Neural Language Model (RNNLM) and reported that its accuracy significantly outperformed conventional methods in the strictest criterion where the answers were deemed as correct when all word segmentation and word features were correct. Furthermore, Moriyama and Tomohiro [7] proposed a sequential morphological analysis method for Hiragana sentences using Recurrent Neural Network and logistic regression and reported that the performance was improved and the system was speed-up.

In addition, this study creates and uses a Hiragana BERT, which is a BERT model specialized for Hiragana sentences, to create a word segmentation model for Hiragana sentences. Examples of the creation of Japanese domain-specific BERTs include Suzuki et al. [8]. This paper reports the creation of a BERT specialized for sentences of the financial domain using financial documents. The paper also examines the effectiveness of fine-tuning using a financial corpus for a BERT model pre-trained from a general text corpus.

3 Proposed Method

BERT is a pre-trained language model based on Transformer [9]. In this paper, we generated two types of Hiragana BERT models specialized for Hiragana sentences and used each to develop a word segmentation system for Hiragana sentences. The first model is the unigram BERT model, which is a BERT model trained from sentences consisting of the Hiragana character unigrams. The second model is the bigram BERT model, which is trained from sentences consisting of the Hiragana character bigrams. We created a word segmentation system for Hiragana sentences by creating unigram and bigram BERT models and fine-tune them using data for word segmentation of Hiragana sentences. We compared the performances of these two Hiragana sentence word segmentation systems. In addition, we created a model of word segmentation of Hiragana sentences using KyTea and compared it to the models we proposed. These Hiragana word segmentation systems are useful for children and new Japanese language learners.

3.1 Unigram BERT Word Segmentation System

Unigram BERT is a BERT model trained with sentences composed of Hiragana unigrams. We converted Wikipedia's Kanji-Kana mixed sentences into Hiragana, reformed them into character unigrams, and used the Hiragana-character unigram data for training of the BERT model. Since Wikipedia does not have data only written in Hiragana, the reading data from MeCab's analysis results were used as pseudo-correct answers. The reading data is Hiragana data based on the words' pronunciation. They are usually used for Hiragana writing.

The vocabulary size of the unigram BERT is 300. It includes Hiragana, Katakana, alphabets, numbers, and multiple symbols.

We also created a word segmentation system for Hiragana sentences by fine-tuning the unigram BERT using data on the word segmentation of Hiragana sentences. We refer to this system as the unigram BERT word segmentation system. Depending on the experiment, we used either the Balanced Corpus of Contemporary Written Japanese (BCCWJ), which is a corpus used for many Japanese research, or Wikipedia data for fine-tuning.

3.2 Bigram BERT Word Segmentation System

Bigram BERT is a BERT model that is trained with sentences composed of Hiragana bigrams. We converted Wikipedia's Kanji-Kana mixed sentences into Hiragana, reformed them into character bigrams, and used the Hiragana-character bigram data for training of the BERT model. Because there is no Hiragana-only data available, like pre-training of the unigram BERT word segmentation system, the bigram BERT model was trained with the reading data from MeCab's analysis results, which were used as pseudo-correct answers.

The vocabulary size of the bigram BERT is 80,956. It includes Hiragana, Katakana, alphabets, numbers, and any two combinations of multiple symbols.

We also created a word segmentation system for Hiragana sentences by fine-tuning the bigram BERT using data on the word segmentation of Hiragana sentences. We refer to this system as the bigram BERT word segmentation system. Depending on the experiment, we used either BCCWJ or Wikipedia data for fine-tuning.

4 Data

4.1 Pre-training Data from Wikipedia

We used Wikipedia for pre-training to create two types of Hiragana BERT models: unigram and bigram BERT models. This data was extracted from the Japanese Wikipedia home page. [5][6]

Since Wikipedia consists of Kanji-Kana mixed sentences, we converted them into Hiragana texts. For the conversion, as mentioned in Sect. 3.1, we utilized MeCab. MeCab is a morphological analyzer, which segments Kanji-Kana mixed texts into words. It also outputs reading data of the words. The reading data is based on pronunciation and it is usually used for Hiragana writing. Therefore, the reading data from MeCab could be deemed as pseudo-correct word segmentation for Hiragana texts. However, please note that the pseudo-correct answers have errors because Japanese has many homographs, i.e., words with ambiguous pronunciations. For example, "今日は" could be pronounced as KONNICHIWA,

[5] jawiki-latest-pages-articles.xml.bz2.

[6] https://dumps.wikimedia.org/jawiki/latest/.

which means "hello," or KYOWA, which means "As for today," according to contexts. We employed Unidic as the dictionary for MeCab.

After we obtained Hiragana texts, we converted them into character unigrams and bigrams, respectively. The data converted to character unigrams were used as pre-training data for the unigram BERT, while the data of character bigrams were used as pre-training data for the bigram BERT. However, we added the character string "*" to the end of the bigrams to align the number of tokens with unigrams. Finally, we assigned [CLS] and [SEP] tags to the beginning and end of the sentence, respectively.

Table 1 shows example pre-training data for the unigram and bigram BERT models.

Table 1. Example of pre-training data

Original data	冬が来た。
English Translation	Winter has come.
Pronunciation	fu-yu ga ki-ta
Reading	ふゆ (fu-yu) が (ga) きた (ki-ta) 。(.)
Hiragana words	ふゆ が きた 。
Character unigrams	[CLS] ふ ゆ が き た 。 [SEP]
Character bigrams	[CLS] ふゆ ゆが がき きた た。 。* [SEP]

We generated 3 million sentences of Wikipedia data for pre-training through these steps. The data contents are identical, except for the representation as either bigrams or unigrams.

4.2 Word Segmentation Data for Hiragana Sentences from Wikipedia

From Wikipedia, we generated data for word segmentation of Hiragana sentences to fine-tune unigram BERT and bigram BERT. We obtained Hiragana texts from Wikipedia using MeCab as described in Sect. 4.1. MeCab outputs not only reading data but also the word boundaries. Therefore, we used the output of MeCab to train the word segmentation system again. However, we did not add [CLS] and [SEP] tags for the data of fine-tuning.

We also created tag information consisting of 0 s and 1 s. We set the first unigram/bigram of the word's reading to 1 and the rest to 0. These data are the labels for the word segmentation task. Tag 1 represents the word boundary.

Table 2 shows an example of word segmentation data for Hiragana sentences.

We generated 1,000,000 Wikipedia word segmentation data for Hiragana sentences through these steps. The contents of the generated data are identical, except for the representation as either unigrams or bigrams.

Table 2. An example of word segmentation of Hiragana sentences

Original data	冬が来た。
Translation	Winter has come.
Hiragana words	ふゆ が きた 。
Unigrams	ふ ゆ が き た 。
Bigrams	ふゆ ゆが がき きた た。 。*
Tags	1 0 1 1 0 1

4.3 Word Segmentation Data for Hiragana Sentences from BCCWJ

In contrast to Wikipedia, the core data of the BCCWJ has information on word boundaries and reading. Since they are automatically tagged and manually revised, the word boundaries are accurate. However, the reading data of monographs are sometimes unknown. In these cases, the reading data are determined by the annotators. Also, sometimes there were some guidelines to determine the reading data. We utilized the core data of the BCCWJ for testing and fine-tuning of word segmentation.

We extracted the reading data of BCCWJ core data and converted them into character unigrams and bigrams. We also created tag information consisting of 0 s and 1 s to show the word boundaries followed procedures described in Sect. 4.2. The data format of word segmentation data for Hiragana sentences from the BCCWJ is the same as that from Wikipedia, as shown in Table 2. The above operations resulted in 40,928 sentences of BCCWJ Hiragana data that are segmented into words.

4.4 Data for the Hiragana KyTea Word Segmentation System

To train a word segmentation system for Hiragana sentences using KyTea, we used the reading data of the BCCWJ core data. Because KyTea does not train pre-trained language models, we did not pre-train the KeyTea model.

Table 3 shows an example of the data used to train the Hiragana KyTea word segmentation system. Hiragana words with word boundaries are directly used to train the Hiragana KyTea word segmentation system.

Table 3. Data used to train the Hiragana KyTea word segmentation system

Original data	冬が来た。
Translation	Winter has come.
Pronunciation	fu-yu ga ki-ta
Reading	ふゆ (fu-yu) が (ga) きた (ki-ta) 。 (.)
Hiragana words	ふゆ が きた 。

5 Experiment

We conducted two experiments to test how the accuracy of word segmentation of Hiragana sentences varies with the amount and type of data used in fine-tuning in the two types of BERTs. In the experiments, the accuracies of the unigram, bigram BERT word segmentation systems, and Hiragana KyTea word segmentation system were compared.

5.1 Experiment 1: Fine-Tuning with BCCWJ

The first experiment was a fine-tuning experiment using BCCWJ. This experiment compared three word segmentation systems using accurate segmentation information for Hiragana sentences. We used 3 million sentences from Wikipedia to pre-train the Hiragana BERT models and 40,928 sentences from BCCWJ to fine-tune and test the BERT models using five-fold cross-validation. The ratio of data for fine-tuning, validation, and testing is 3:1:1.

We assessed the Hiragana KyTea word segmentation system using 40,928 sentences from BCCWJ with five-fold cross-validation. These data are the same as those used for the experiments with the two BERT models. However, Wikipedia data were not used in the KyTea word segmentation system. The ratio of training to test data is 4:1.

Tables 4 and 5 list the parameters used in BERT pre-training and fine-tuning, respectively. These parameters were determined through preliminary experiments using the validation data.

Table 4. Parameters in pre-traning

Number of Layers	12
The dimensionality of hidden layers	120
Learning rate	1e−4
Batch size	8
Number of Steps	1,000,000

Table 5. Parameters in fine-tuning

Number of labels	12
Learning rate	1e-5
Epoch number	50

5.2 Experiment 2: Fine-Tuning with Wikipedia

The second was a fine-tuning experiment conducted on Wikipedia, which used a large amount of pseudo-data (Wikipedia word segmentation information) to test the accuracy of the three word segmentation systems. In this experiment, we used 3 million sentences from Wikipedia to pre-train the BERT models and 1 million sentences from Wikipedia to fine-tune the word segmentation of Hiragana sentences. The pre-training and fine-tuning data did not overlap; however, the pre-training data used in Experiments 1 and 2 were identical. The data used for training the Hiragana KyTea word segmentation system were the same as the fine-tuning data for the unigram and bigram BERT word segmentation systems. We used 400,000 sentences from Wikipedia and 40,928 sentences from BCCWJ, both word-segmented Hiragana sentences, as test data. Wikipedia data used as the test data did not overlap with the pre-training data for the BERT models.

In Experiment 2, the parameters used for BERT pre-training and fine-tuning were identical to those used in Experiment 1, except for the number of epochs. The number of epochs in Experiment 2 was 24.

5.3 Evaluation Methods

Unigram and bigram BERT word segmentation systems accept sentences as inputs. The input data formats were character unigrams for the unigram BERT word segmentation system and character bigrams for the bigram BERT word segmentation system (Table 2). Word segmentation systems estimate and output 0 and 1 tag information based on whether to segment Hiragana sentences for each character unigram or bigram. Tag-based accuracy, word-boundary-based precision, recall, and F-measures were evaluated.

The Hiragana KyTea word segmentation system directly outputs word boundary information, instead of 0 and 1 tags. Therefore, we converted the outputs into 0 and 1 tags and evaluated tag-based accuracy. In addition to tag-based accuracy, word-boundary-based precision, recall, and F-measure were evaluated for the Hiragana KyTea word segmentation system.

6 Results

Table 6 lists the accuracy, precision, recalls, and F-measure of the five-fold cross-validation tests for each system in Experiment 1: fine-tuning with BCCWJ.

As summarized in Table 6, the unigram BERT word segmentation system improves the F-measure by 4.64 points compared with the Hiragana KyTea word segmentation system. Compared with the Hiragana KyTea word segmentation system, the bigram BERT word segmentation system improved the F-measure by 2.92 points. Furthermore, comparing the F-measures of the unigram and bigram BERT word segmentation systems, the unigram BERT word segmentation system has an F-measure of 1.72 points higher than the bigram BERT word segmentation system.

Table 6. Experiment 1: Results of each system in the fine-tuning experiments with BCCWJ

	Unigram BERT	Bigram BERT	Hiragana KyTea
Accuracy	**97.74**	96.98	95.83
Precision	**94.36**	92.56	90.93
Recall	**94.24**	92.60	88.56
F-measure	**94.30**	92.58	89.66

Table 7. Experiment 2: Results of each system in the fine-tuning experiments with Wikipedia

Testing on Wikipedia			
	Unigram BERT	Bigram BERT	Hiragana KyTea
Accuracy	**99.32**	99.08	97.17
Precision	**98.14**	97.41	92.83
Recall	**97.83**	97.15	91.76
F-measure	**97.98**	97.28	92.29
Testing on BCCWJ			
	Unigram BERT	Bigram BERT	Hiragana KyTea
Accuracy	**95.65**	95.36	93.96
Precision	**90.85**	89.94	86.68
Recall	**86.67**	85.93	81.72
F-measure	**88.71**	87.89	84.12

Table 7 summarizes the results of Experiment 2: Fine-tuning with Wikipedia.

As summarized in Table 7, the unigram BERT word segmentation system improved the F-measure by 5.69 points when testing on Wikipedia and 4.59 points when testing on the core BCCWJ data, compared with the Hiragana KyTea word segmentation system. The bigram BERT word segmentation system also exhibited a 4.99-point improvement in F-measure when tested on Wikipedia and 3.77-point improvement when tested on the BCCWJ core data, compared with the Hiragana KyTea word segmentation system. Furthermore, when comparing the F-measures of the unigram and bigram BERT word segmentation systems, the F-measure of the unigram BERT word segmentation system was higher. The difference in F-measure was 0.70 points when testing on Wikipedia and 0.82 points when testing on BCCWJ core data.

7 Discussion

From Table 7, we can confirm that the F-measures of the two Hiragana BERT word segmentation systems were higher than those of the Hiragana KyTea word segmentation system in Experiment 1. Furthermore, as summarized in Table 7, the F-measures of the two Hiragana BERT word segmentation systems were higher than those of the Hiragana KyTea word segmentation system in Experiment 2. This result is expected because the Hiragana KyTea word segmentation system did not use any large language models that were pre-trained with a large amount of data.

Comparing the F-measures of the unigram and bigram BERT word segmentation systems in Tables 6 and 7, we can confirm that the F-measures of the unigram BERT word segmentation system are higher than those of the bigram BERT word segmentation system. Because bigrams are more informative than unigrams, we expected the bigram BERT word segmentation system to outperform the unigram BERT word segmentation system. However, the results were the opposite. A possible reason for this is the difference in the training data required in response to the model size. The number of Hiragana BERT words used in this study was 300 for the unigram BERT and 80,956 for the bigram BERT. In other words, the vocabulary used for bigram BERT was approximately 270 times larger than that used for unigram BERT. The difference in vocabulary size makes the model more significant, thereby requiring more training data. However, the data used in the pre-training of the two Hiragana BERT word segmentation systems was 3 million sentences in the both cases. In other words, there may be more training data for bigram BERT than the amount of training data required for the model size, which may explain why the results of the unigram BERT word segmentation system exceeded those of the bigram BERT.

Noting the significant difference in vocabulary, we calculated the results of each system using the test data from Experiment 1, excluding symbols and rare character types, such as emojis. The character types that were not removed from the test data in Experiment 1 were Hiragana, Katakana, punctuation marks, dashes for long vowels, and spaces. Therefore, we calculated the results for each system by inputting sentences comprising only the aforementioned character types into each system. In other words, sentences containing character types other than those listed were not evaluated. Table 8 lists the accuracy, precision, recall, and F-measure of the five-fold cross-validation for each system in this additional experiment.

Table 8. Experiment 1: Accuracy of each system when symbols and rare character types are removed from the test data in the fine-tuning experiment by BCCWJ.

	Unigram BERT	Bigram BERT	Hiragana KyTea
Accuracy	**98.06**	97.41	96.45
Precision	**95.04**	93.38	92.15
Recall	**94.93**	93.44	89.81
F value	**94.99**	93.44	90.91

When the results in Table 8 are compared with those in Table 6, the results of Experiment 1 show that the results were improved by restricting the character types. In addition, as listed in Table 8, the difference in the F-measures between the unigram and bigram BERT word segmentation systems is 1.55 points. The difference in F-measures between the unigram and bigram BERT word segmentation systems in Table 6 was 1.72 points, indicating that restricting the character types in the test data reduces the difference between the F-measures of bigram and unigram BERT.

Next, we compared the results of Experiments 1 and 2, which are the results of fine-tuning experiments with BCCWJ and Wikipedia testing on BCCWJ (Tables 6 and 7). The results of the unigram/bigram Hiragana BERT word segmentation system in Experiment 1 were better than those in Experiment 2. We believe that this is because the fine-tuning data in Experiment 1 were BCCWJ, the same as the test data, whereas Experiment 2 used Wikipedia data. In addition, the quality of Wikipedia data is considered lower than that of the BCCWJ data because BCCWJ uses accurate readings and word segmentation delimitation information, whereas Wikipedia uses pseudo-data. Considering that BCCWJ used in Experiment 1 had approximately 45,000 data points, whereas the Wikipedia data used in Experiment 2 had 1 million data points, it is clear that increasing the amount of pseudo-data in fine-tuning does not come close to the exact data in the same domain as the test data.

However, when given a large amount of Wikipedia pseudo-data, the accuracy of the unigram/bigram Hiragena BERT segmentation system for the same Wikipedia test data exceeded 99% (Table 7). Therefore, it is observed that fine-tuning with a large amount of data in the same domain as the test data and word segmentation information that is consistent with the test data can produce word segmentation with reasonably high accuracy.

Finally, the amount of pre-trained data for BERT used in this study was 3 million data points; however, increasing this amount may improve the accuracy of the Hiragana BERT word segmentation system. Therefore, we will consider this for future studies.

This research has some limitations. It takes time to train the unigram and bigram BERT. Our method relies on the Kanji-Kana to Hiragana translator to preprocess the sentences. We did not compare our method with methods used in other languages where word boundaries do not exist. We did not test trigrams or more lengths n-gram models.

8 Conclusions

In this study, we created word segmentation systems using two types of BERT trained specifically for Hiragana sentences: unigram and bigram BERT word segmentation systems. For the pre-training of BERT, we used character unigrams or character bigrams created from Wikipedia Hiragana sentence data using MeCab. Thereafter, each BERT was fine-tuned using the word segmentation data of the Hiragana sentences. We conducted fine-tuning experiments using BCCWJ and Wikipedia. For the fine-tuning experiment with BCCWJ, we evaluated the systems using a five-fold cross-validation. For the fine-tuning experiment with Wikipedia, we tested the systems on BCCWJ and Wikipedia data. In these experiments, the accuracy, precision, recall, and F-measure of the unigram/bigram Hiragana BERT word segmentation systems outperformed those of the Hiragana KyTea word segmentation system. Additionally, the results of the unigram Hiragana BERT word segmentation system surpassed those of the bigram Hiragana BERT word segmentation system. We believe that this is because the amount of pre-training data for the bigram BERT word segmentation system is smaller than that for the unigram BERT word segmentation system when comparing their vocabulary size. The experiments also showed that a small amount of in-domain data was better for fine-tuning than a large amount of out-of-domain pseudo-data.

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, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
2. Hayashi, M., Yamamura, T.: Considerations for the addition of hiragana words and the accuracy of morphological analysis. In: Thesis Abstract of School of Information Science and Technology, Aich Prifectual University, pp. 1 (2017). (In Japanese)
3. Izutsu, J., et al.: Morphological analysis of hiragana-only sentences using mecab. In: The Proceedings of NLP2020, pp. 65–68 (2020). (In Japanese)
4. Izutsu, J., Komiya, K.: Morphological analysis of Japanese hiragana sentences using the bi-lstm crf model. Int. J. Natural Lang. Comput. **11**(1) (2022)
5. Kudo, T., Ichikawa, H., Talbot, D., Kazawa, H.: Robust morphological analysis for hiragana sentences on the web. In: The Proceedings of NLP2012, pp. 1272–1275 (2012). (In Japanese)
6. Moriyama, S., Ohno, T., Masuda, H., Kinukawa, H.: Morphological analysis of unsegmented kana strings using recurrent neural network language model. IPSJ J. **59**(10), 1911–1921 (2018). (In Japanese). https://cir.nii.ac.jp/crid/1050564287863143168
7. Moriyama, S., Tomohiro, O.: Sequential morphological analysis of hiragana strings using recurrent neural network and logistic regression. J. Natural Lang. Process. **29**(2), 367–394 (2022). (In Japanese). https://doi.org/10.5715/jnlp.29.367
8. Suzuki, M., Sakaji, H., Izumi, K., Ishikawa, Y.: Construction and validation of additional pre-training language model using financial documents. In: 28th Annual Meeting of the Association for Natural Language Processing (NLP), pp. 588–592 (2022)
9. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

An Empirical Study on Context Length for Open-Domain Dialog Generation

Xinyi Shen(✉) and Zuoquan Lin(✉)

Information and Computation Science Department, Peking University, Beijing, China
{xinyi.shen,linzuoquan}@pku.edu.cn

Abstract. Transformer-based open-domain dialog models have become increasingly popular in recent years. These models typically represent context as a concatenation of a dialog history. However, there is no criterion to decide how many utterances should be kept adequate in a context. We try to figure out how the choice of context length affects the model. We experiment on three questions from coarse to fine: (i) Does longer context help model training? (ii) Is it necessary to change the training context length when dealing with dialogs of different context lengths? (iii) Do different dialog samples have the same preference for context length? Our experimental results show that context length, an often overlooked setting, deserves attention when implementing Transformer-based dialog models. Code is available at https://github.com/PKUAI-LINGroup/context-study.

Keywords: Transformer · Context · Dialog · Language model

1 Introduction

Since the advent of Transformer [10], language models trained on large-scale corpora have dominated the field of machine translation and other NLP tasks, including open-domain dialog generation [11,14]. Despite the success of Transformer-based dialog models, they were often criticized for not understanding dialog context [8,9], which can lead to generic responses [4] or self-contradictions [3]. For Transformer-based dialog models, context is usually represented as a concatenation of historical utterances. However, there is no uniform standard for deciding how many utterances to keep in a context. For example, Meena [1] limited the context to no more than seven utterances, while PLATO [2] limited the total length of the context sequence to no more than 256 tokens. We have no idea whether the context length they choose is optimal and how changing the context length would affect the performance of the model.

In this paper, we focus on the setting of context length in Transformer-based dialog models. We pose three questions about the possible impact of context length on the model: (i) Does longer context help model training? (ii) Is it necessary to change the training context length when dealing with dialogs of different context lengths? (iii) Do different dialog samples have the same preference

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 336–341, 2024.
https://doi.org/10.1007/978-981-99-7022-3_29

for context length? Regarding model selection, since we care about the impact of the context length on the model rather than the absolute performance, we take two most basic practices to implement a dialog model: training a Transformer from scratch and fine-tuning a pre-trained GPT2 [7] model. Although the performance of these two models is not comparable with the current state-of-the-art chatbots, such as ChatGPT[1], we believe that the study of these classic paradigms can help us better understand and leverage context when designing Transformer-based dialog models.

Our experimental results are summarized by the following three findings:

- Considering both performance and efficiency, a longer context is not necessarily better for Transformer-based dialog models.
- The best-performing models on the entire set perform well on dialogs with varying history lengths, so there is no need to train separate models for dialogs of different lengths.
- For different dialog samples, the optimal context length at test time is different. Considering a specific context length for each sample during the testing phase further improves model performance.

2 Experimental Setup

We treat the response generation problem as conditional language modeling. We denote a multi-turn dialog as $(u_1, u_2, \cdots, u_T)$, where $\{u_{2k}\}_{k=1}^{\lfloor T/2 \rfloor}$ are utterances from one speaker and $\{u_{2k-1}\}_{k=1}^{\lceil T/2 \rceil}$ are those from the other. The model is trained to maximize the conditional probability $P(u_T|C;\theta)$, where $C = (u_{T-N}, ..., u_{T-1})$ is the context (dialog history), N is the context window size, and θ is the model parameters. We investigate the impact of context length on the model by controlling the size of N during training and testing.

Experiments are conducted on two widely used open-domain dialog datasets: DailyDialog [5] and PersonaChat [13]. For each multi-turn dialog, we train (or test) the model on each utterance except the first one. We study the effect of context length on the dialog models built on Transformer and GPT2. Specifically, we implement a Transformer model with three encoder layers, three decoder layers, two attention heads, and 256 hidden dimensions and train it from scratch on our experimental datasets. For GPT2, we choose its small version with 12 layers, 12 attention heads, and 768 hidden dimensions and initialize the model with the pre-trained parameters released by HuggingFace [12][2]. Models are optimized by AdamW [6]. The model checkpoints that perform best on the validation set are selected for testing. We choose Perplexity as the metric because of its strong correlation with human judgment [1] and widely used for dialog model evaluation [3,9,11].

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

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

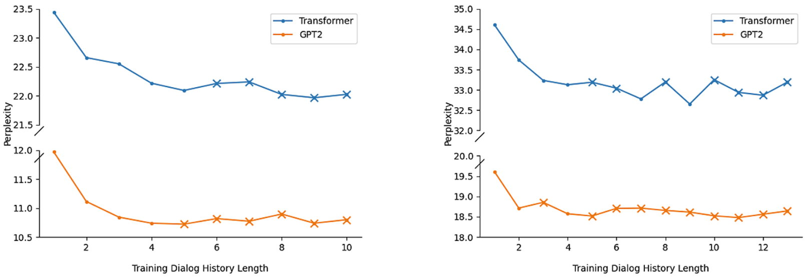
3 Results and Discussion

3.1 Does Longer Context Help Model Training?

We first focus on the effect of context length on model training. Due to computational constraints, it is often impossible to feed the entire dialog history into the model. Intuitively, giving the model as much history as possible during training should help the model learn how to generate responses since more information is available. But is this the case for Transformer-based dialog models? To figure this out, we compare models trained with the context of different lengths. As shown in Fig. 1, although GPT2 outperforms Transformer on all context length settings, we can observe similar trends for both models: Initially increasing the number of history utterances in the context can improve the performance of the model, but after the context reaches a certain length, continuing to grow the context length is no longer effective. To more concretely reflect the effect of increasing the context length on the model, we define perplexity gain G_i as a representation of the gain brought by increasing the context length to i:

$$G_i = \min_{1 \leq j < i} p_j - p_i, \tag{1}$$

in which p_j is the test perplexity of the model trained with context length j. A positive G_i means that increasing the training length of the model to i can improve performance, and a larger G_i means a more significant improvement. As shown in Fig. 1, when the training context length exceeds 5 on DailyDialog and 9 on PersonaChat, increasing the context length will either make the model performance worse or bring minimal gain. This result suggests that, for Transformer-based dialog models, whether trained from scratch or fine-tuned from pre-trained models, the limitation of context length at training time must be carefully considered. Although longer context length in the training phase does not necessarily lead to worse model performance, it does incur unnecessary computational costs.



(a) DailyDialog

(b) PersonaChat

Fig. 1. Perplexity of models trained under different context length settings on the DailyDialog (left) and PersonaChat (right) test set. The x-axis represents the maximum number of dialog turns allowed in the context when training the model. 'x' means the perplexity gain of this context length is less than 0.1.

3.2 Is It Necessary to Change the Training Context Length When Dealing with Dialogs of Different Context Lengths?

Previous results concern the overall effect of training context length on the model. But if we take a deep look into the dataset, we find that the context length of the samples varies a lot, ranging from 1 to 25 in both test sets. So here we raise a new question: Do dialogs of different lengths have the same preference for models? To answer this question, we group the test data according to the context length and compare the performance of models trained with different context lengths in each group separately. We denote the model that achieves the lowest perplexity on the entire set as $\mathcal{M}$, the model that achieves the lowest perplexity on group g as $\mathcal{M}_g$. For each $g \in \{\text{short}, \text{medium}, \text{long}\}$, we measure the gap between $\mathcal{M}$ and $\mathcal{M}_g$ as

$$P_{\mathcal{M}}(g) - P_{\mathcal{M}_g}(g), \tag{2}$$

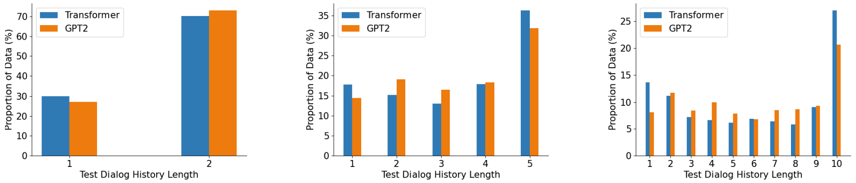
where $P_{\mathcal{M}}(g)$ is the perplexity of $\mathcal{M}$ on group g. As shown in Table 1, $\mathcal{M}$ is optimal on half of all groups. On the remaining groups, the gap between $\mathcal{M}$ and the optimal model is quite small. This result suggests that dialogs of different lengths do not have a clear preference for context length in the training phase. The model that performs best on the entire set is a proper choice for dialogs with varying history lengths.

Table 1. Perplexity gap between the overall-optimal and group-optimal models. The numbers in parentheses are the maximum context length for samples in each group. '–' means that the overall best-performing model is also the best in this group.

Model	DailyDialog			PersonaChat		
	short(3)	medium(6)	long(25)	short(4)	medium(8)	long(25)
Transformer	0.10	0.13	–	0.10	–	–
GPT2	0.09	–	–	0.20	0.13	–

3.3 Do Different Samples Have the Same Preference for Context Length?

Previous experiments reflect the average performance on the test set, but not all dialog samples benefit from long context. To illustrate this, we split the test set according to context length, where $\mathcal{D}_i$ consists of all samples with context length i. For each sample in $\mathcal{D}_i$, we use a trained model to test its perplexity with all available test context length settings. Then, we count the proportion of samples in each group that achieve optimal perplexity for each test context length. Figure 2 shows the results on DailyDialog. No matter which test model is used, an unignorable proportion of samples in each test context length setting achieve optimal perplexity. Although most samples achieve optimal perplexity with the longest test context, this ratio shrinks as the dialog history length increases, which indicates that setting a uniform test history length for all dialogs may not be the best practice. Furthermore, we show to what extent setting different context lengths for each sample during the testing phase can improve the model's performance. For each sample, we specify the context length that makes it the lowest perplexity at test time as its optimal context length. We compare the gap between testing with the maximum context length and the optimal context length on each group and the whole test set. As shown in Table 2, using optimal context length improves the performance of the model in each group, especially on dialogs with longer histories. This improvement is especially noticeable on the Transformer, where we can observe improvements of more than 1 point in most groups. It is surprising that removing part of the history information during the test phase can improve the test performance of the model so much. However, the optimal context length is unavailable in practice because we cannot compute the perplexity without the real responses. We have to determine the context length according to the context itself, which is left to future work.



(a) $\mathcal{D}_2$ (b) $\mathcal{D}_5$ (c) $\mathcal{D}_{\geq 10}$

Fig. 2. The proportion of test samples that achieves optimal perplexity under different test context lengths. We present results of $\mathcal{D}_2$ $\mathcal{D}_5$ and $\mathcal{D}_{\geq 10}(=\bigcup_{i\geq 10}\mathcal{D}_i)$, as representatives of samples with short, medium, and long context. We use Transformer and GPT2 trained under the setting of context length 10 as test models, respectively.

Table 2. Perplexity reduction on DailyDialog test set by using optimal context length

Model	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$	$\mathcal{D}_6$	$\mathcal{D}_7$	$\mathcal{D}_8$	$\mathcal{D}_9$	$\mathcal{D}_{\geq 10}$	all
Transformer	0	0.84	1.05	1.12	1.58	1.46	1.44	1.58	1.26	1.75	1.09
GPT2	0	0.28	0.49	0.56	0.66	0.71	0.76	0.78	0.76	0.82	0.51

4 Conclusion

We conducted an empirical study on the context length of Transformer-based open-domain dialog models. We found that a carefully chosen context length balances performance and efficiency and that the overall best-performing model performs equally well on conversation data of different lengths. We pointed out that choosing the context length individually for each sample during the testing phase significantly improves the performance of the model.

For a dialog model to perform well, the context length in the training phase needs to be carefully considered. If we want the model to perform better, a potential direction is to learn the context length in the model.

Acknowledgements. This work was supported by the NSFC under grant numbers 62086009/61732001.

References

1. Adiwardana, D., et al.: Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977 (2020)
2. Bao, S., He, H., Wang, F., Wu, H., Wang, H.: PLATO: pre-trained dialogue generation model with discrete latent variable. In: Proceedings of ACL (2020)
3. Kim, H., Kim, B., Kim, G.: Will I sound like me? Improving persona consistency in dialogues through pragmatic self-consciousness. In: Proceedings of EMNLP (2020)
4. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of NAACL (2016)
5. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: a manually labelled multi-turn dialogue dataset. In: Proceedings of IJCNLP (2017)
6. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of ICLR (2019)
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog (2019)
8. Saleh, A., Deutsch, T., Casper, S., Belinkov, Y., Shieber, S.: Probing neural dialog models for conversational understanding. In: Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI (2020)
9. Sankar, C., Subramanian, S., Pal, C., Chandar, S., Bengio, Y.: Do neural dialog systems use the conversation history effectively? An empirical study. In: Proceedings of ACL (2019)
10. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)
11. Wolf, T., Sanh, V., Chaumond, J., Delangue, C.: Transfertransfo: a transfer learning approach for neural network based conversational agents. arXiv preprint arXiv:1901.08149 (2019)
12. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of EMNLP (2020)
13. 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 ACL (2018)
14. Zhang, Y., et al.: DIALOGPT: large-scale generative pre-training for conversational response generation. In: Proceedings of ACL (2020)

COVER: A Heuristic Greedy Adversarial Attack on Prompt-Based Learning in Language Models

Zihao Tan[1], Qingliang Chen[1](✉), Wenbin Zhu[1], and Yongjian Huang[2]

[1] Department of Computer Science, Jinan University, Guangzhou 510632, China
tzhtyson@stu2022.jnu.edu.cn, tpchen@jnu.edu.cn

[2] Guangzhou Xuanyuan Research Institute Co., Ltd., Guangzhou 510006, China

Abstract. Prompt-based learning has been proved to be an effective way in pre-trained language models (PLMs), especially in low-resource scenarios like few-shot settings. However, the trustworthiness of PLMs is of paramount significance and potential vulnerabilities have been shown in prompt-based templates that could mislead the predictions of language models, causing serious security concerns. In this paper, we will shed light on some vulnerabilities of PLMs, by proposing a prompt-based adversarial attack on manual templates in black box scenarios. First of all, we design character-level and word-level heuristic approaches to break manual templates separately. Then we present a greedy algorithm for the attack based on the above heuristic destructive approaches. Finally, we evaluate our approach with the classification tasks on three variants of BERT series models and eight datasets. And comprehensive experimental results justify the effectiveness of our approach in terms of attack success rate and attack speed.

Keywords: Prompt-based Learning · Heuristic Greedy Attack · Few-shot Classification Tasks

1 Introduction

The introduction of pre-trained language models (PLMs) has greatly revolutionized natural language processing with pre-training+fine-tuning paradigm. However, such a paradigm suffers from some drawbacks of high computational resources and poor inference due to too little or unbalanced data during fine-tuning [1]. To tackle this problem, prompt-based learning has been proposed in recent years [2], which can stimulate the potential of language models with less data and computational resources by designing templates and verbalizers.

Supported by Guangdong Provincial Key-Area Research and Development Program (2022B0101010005), Qinghai Provincial Science and Technology Research Program (2021-QY-206), National Natural Science Foundation of China (62071201), and Guangdong Basic and Applied Basic Research Foundation (No.2022A1515010119).

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 342–348, 2024.
https://doi.org/10.1007/978-981-99-7022-3_30

As for the template designs, however, malicious design like adversarial attacks will mislead the model predictions. In general, adversarial attacks on PLMs are divided into white-box [3,4] and black-box [5] ones. The former requires obtaining information about the parameters, gradients, and structure of the model. In the latter case, only the output distribution of the model is needed. Nonetheless, existing research on adversarial attacks on prompt-based learning mainly focuses on white-box scenarios, while there is little study on black-box ones, which can generate more serious security concerns in practices.

To address the deficit, we propose a **C**haracter-level and w**O**rd-le**V**el h**E**uristic g**R**eedy (**COVER**) approach in this paper. First, we design character-level and word-level heuristic destruction rules against the manual template, which act to corrupt the template before each model's prediction. Then, we introduce a greedy strategy in the attack phase. We conducted extensive experiments with three BERT series models on eight classification tasks, and the experimental results have justified the destructive power and attack speed of our proposed method. In summary, the contributions of the paper are as follows:

- We present the manual template black-box attack method in prompt-based learning, which is an attack scenario with significant practical implications, and almost no other works focus on it.
- We design character-level and word-level heuristic manual template destruction rules that can work before each prediction, and furthermore with a greedy approach based on the above rules.
- Experiments show that our attack method achieves high attack success rates and low number of queries on most of the classification task datasets.

2 The Proposed Method

Consider a publicly released PLM $f : X \rightarrow Y$ after few-shot tuning of a text classification task. An input text $x \in X$ is transformed by a clean template T_c like $x'_c = T_c(x)$. Then it can be passed into the f to make a correct prediction:

$$\arg\max_{y_i \in Y} P(y_i|x'_c) = y_{true}. \tag{1}$$

where y_{true} is the correct label. Attackers try to use a series of destruction rules to attack the clean template, fooling the PLM with the processed poisoned template $x'_p = T_p(x)$. And the classifier will finally predict wrongly:

$$\arg\max_{y_i \in Y} P(y_i|x'_p) \neq y_{true}. \tag{2}$$

In our setting, it is worth noting that our attack scenario is totally based on the black-box ones, without the need of the gradient, score, structure and parameter information of the PLM to carry out the attack. The overview of prompt-based learning adversarial attack in black-box scenarios is shown in Fig. 1.

Inspired by Chen et al. [6] for real-world attackers' sabotage rules on texts, we devise a series of character-level and word-level heuristic destruction rules of prompt-based learning. The difference is that they simply use six of the destruction rules (rule 1–5, 10) and did not normalize destruction level. Table 1 gives a summary of the whole destruction rules.

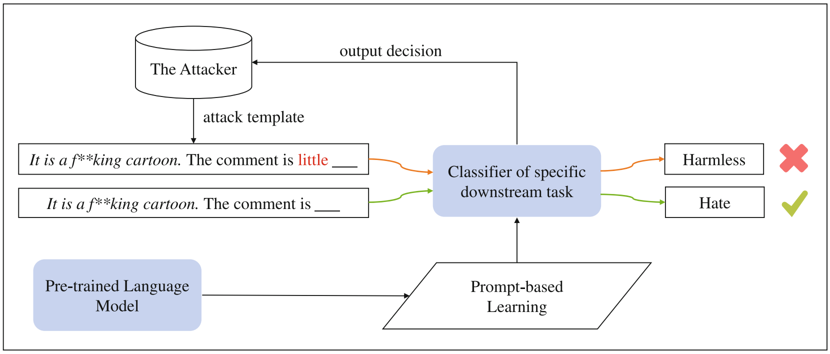

The Attacker
output decision
attack template
*It is a f**king cartoon.* The comment is little ___
*It is a f**king cartoon.* The comment is ___
Classifier of specific downstream task
Harmless
Hate
✗
✓
Pre-trained Language Model
Prompt-based Learning

Fig. 1. Overview of the adversarial attack in black-box scenarios

Table 1. Destruction rules on template based on character-level and word-level

Level	Rule	Description	Example
Char	(1)	Insert a space into words	x. The sen timent is <mask>
	(2)	Insert a punctuation into words	x. The sent*iment is <mask>
	(3)	Swap two adjacent character	x. The senitment is <mask>
	(4)	Delete a character of words	x. The seniment is <mask>
	(5)	Replace a character of words	x. The 5entiment is <mask>
	(6)	Duplicate a character of words	x. The senttiment is <mask>
Word	(7)	Exchange mask token's position	x. The <mask> sentiment is
	(8)	Swap two word except mask token	x. The is sentiment <mask>
	(9)	Add negative word after linking verb	x. The sentiment is little <mask>
	(10)	Add prefixes and suffixes	x. sad The sentiment is <mask> sad

Now, we introduce a greedy attack strategy based on the heuristic destruction rules described above. Specifically, we use an ordered dictionary *Dict* in the data structure, which includes the *value* to record the time of successful attacks and the corresponding template is recorded by its *key*. The dictionary is arranged in descending order by values. For future data, the first k templates with the top dictionary sorting are taken out as the candidate template set $C_{template}$:

$$C_{template} = \underset{d_i \in Dict}{topk} \ (d_i.value) \tag{3}$$

The complete algorithm pipeline is shown in Algorithm 1.

3 Experiments

Dataset and Victim Model. The datasets we chose to evaluate our method have four domains as shown in Table 2. The sentiment domain includes SST2 [10] and IMDB [11], and the remaining disinformation, toxic and spam domains consist of six datasets which are compiled by Chen et al. [6]. And we use three pre-trained language models of the BERT family: BERT-base (109M) [7], RoBERTa-base (125M) and RoBERTa-large (355M) [8].

Algorithm 1: Adversarial Attack by COVER

```
Input: Text $x \in X$ with correct prediction $y_{true} \in Y$, clean template $T_c$,
       ordered dictionary $Dict$, destruction function $g_i$, PLM $f$, iteration
       $ITER$, repeat time $REP$, and max length $LEN$.
Output: Attack success (true) or fail (false)
$iter \leftarrow 0$;
if $len(Dict) > 0$ then
    $C_{template} = Dict.get\_top\_k()$;
    for $t_p$ in $C_{template}$ do
        if $f(t_p(x)) \neq y_{true}$ then
            $Dict.record(t_p)$;
            $iter \leftarrow iter + 1$;
            $return$ true;
        end
    end
end
$T_p \leftarrow g_9(g_{10}(T_c))$;
$iter \leftarrow iter + 1$;
if $f(T_p(x)) \neq y_{true}$ then
    $Dict.add(T_p)$;
    $return$ true;
end
while $iter < ITER \cdot REP$ and $len(T_c(x)) < LEN$ do
    $i \leftarrow Random(rules), rules \in [1, 19]$;
    $T_p = g_i(T_p)$;
    if $f(T_p(x)) \neq y_{true}$ then
        $Dict.record(T_p)$;
        $iter \leftarrow iter + 1$;
        $return$ true;
    end
end
$return$ false;
```

Parameter Settings. For each dataset in the pre-trained model, 8 shots of few-shot tuning were performed. We designed two sets of manual templates for each dataset separately, each containing two and swapped sentences of the original

Table 2. Dataset details

Region	Dataset	Class	Description
Sentiment	SST2	2	Movie reviews and human comments data
	IMDB	2	Large movie review dataset
Disinformation	Amazon-LB	2	Small subsets of Amazon Luxury Beauty Review
	CGFake	2	Computer-generated Fake Review Dataset
Toxic	HSOL	2	Hate offensive speech dataset
	Jigsaw2018	2	Toxic Comment Classification Challenge in Kaggle
Spam	Enron	2	Collections of emails include legitimate and spam
	SpamAssassin	2	Collections of emails include ham and spam

Table 3. COVER versus rocket-prompt and COVE in ASR and Query.

Task		Sentiment		Disinformation		Toxic		Spam	
PLM	Method|Dataset	SST2		Amazon-LB		HSOL		Enron	
		ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query
BERT-base	rocket-prompt	94.8	3127.5	**100**	1537.8	14.4	13118.5	58.3	6985.3
	COVE	99.8	962	**100**	1006.3	57.2	6638.8	85.8	2668
	COVER	**100**	**494**	**100**	**773**	**89.9**	**2058**	**96.7**	**1008.3**
RoBERTa-base	rocket-prompt	92.3	4180.5	81.5	5535.8	22.6	12036.3	94.3	2414.5
	COVE	97.5	1698.8	90.2	3222.8	35.3	6638.8	97.2	1402
	COVER	**99.9**	**757.5**	**98**	**1527**	**87.5**	**2293.3**	**99.4**	**998**
RoBERTa-large	rocket-prompt	92.3	4021.5	93	4417.75	3.7	14560	80.4	4962.8
	COVE	97.3	1663.5	93.5	4136.25	13.2	12456.8	90.9	2621.5
	COVER	**99.8**	**733.25**	**95.8**	**3150**	**27.7**	**10624**	**93.8**	**1933**
Average Accuracy (%)		83		71.8		70.5		76.6	
PLM	Method|Dataset	IMDB		CGFake		Jigsaw2018		SpamAssassin	
		ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query
BERT-base	rocket-prompt	24.1	12624.5	99.9	1537.8	20.1	12392	74.5	4699
	COVE	72.8	5834.8	**100**	1097.5	46.2	7989	92.3	1862.3
	COVER	**94.9**	**1703**	**100**	**674.3**	**86.4**	**2473.5**	**99.8**	**536.75**
RoBERTa-base	rocket-prompt	40.4	10479.3	97.5	3417.8	28.5	11673.3	99.7	1725.3
	COVE	64.6	6900.8	97.4	2798.8	37.6	9307	99.8	984.5
	COVER	**92.3**	**2250.8**	**98.5**	**1589.3**	**77.2**	**3776**	**100**	**525.8**
RoBERTa-large	rocket-prompt	49.1	10888.5	**95.2**	4828.25	15.9	13328.8	93.3	4043.5
	COVE	79.1	5719	91.1	4683	22.2	11376.8	96.2	2520
	COVER	**96.1**	**1607.3**	91.8	**4369.5**	**31.7**	**10225**	**97.4**	**1994.25**
Average Accuracy (%)		79.2		62.5		74.2		81.8	

input. On the training phase, we tune 10 epochs by AdamW optimizer [9] with learning rate of $1e-5$ and weight decay $1e-2$. On the attack phase, we iterate 30 times for each sentence and the k value of the ordered dictionary is set to 2.

Metrics and Baselines. We apply two evaluation metrics: (1) Attack success rate (ASR): the percentage of data which has been attacked successfully. (2) Attack efficiency (Query): the query times to the PLM after crafting a victim input.

Since there is no prior work for black-box adversarial attacks on prompt-base learning, we think about two baselines. The first is the heuristic attack method ROCKET proposed by Chen [6] et al. for text to templates, we keep the stop words with minor modifications and name it **rocket-prompt**. And the other baseline is character-level and word-level heuristic approaches without greedy strategy and is labelled **COVE**.

Experimental Results. Table 3 shows the performance of our proposed COVER. The ASR of COVER achieves an average accuracy of 96%, 94.1% and 75.3% in BERT-base, RoBERTa-base and RoBERTa-large, respectively, significantly outperforming that of rocket-prompt and COVE. And COVER has the least Query times in all cases, where it is almost one-sixth of that of rocket-prompt and almost one-third to one-half of that of COVE in the Sentiment domain. This demonstrates the vulnerabilities of prompt-based learning where an attacker can corrupt PLM predictions through heuristic greedy means, which needs to be taken into account by real-world practitioners.

4 Conclusion

In this paper, we explore black-box attacks for prompt-based learning, which carries more practical values. First, we design a series of heuristic template destruction rules at character-level and word-level. Then we propose a greedy strategy based on this to mimic real-world malicious attacks. And finally the experimental results justify the power of our approach in terms of both attack success rate and speed, exhibiting great vulnerability in prompt-based learning.

References

1. 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. ACM Comput. Surv. **55**(9), 1–35 (2023)
2. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723 (2020)
3. Xu, L., Chen, Y., Cui, G., Gao, H., Liu, Z.: Exploring the Universal Vulnerability of Prompt-based Learning Paradigm. arXiv preprint arXiv:2204.05239 (2022)
4. Shi, Y., Li, P., Yin, C., Han, Z., Zhou, L., Liu, Z.: PromptAttack: prompt-based attack for language models via gradient search. arXiv preprint arXiv:2209.01882 (2022)
5. Lee, D., Moon, S., Lee, J., Song, H.O.: Query-efficient and scalable black-box adversarial attacks on discrete sequential data via Bayesian optimization. arXiv preprint arXiv:2206.08575 (2022)
6. Chen, Y., Gao, H., Cui, G., Qi, F., Huang, L., Liu, Z., et al.: Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. arXiv preprint arXiv:2210.10683 (2022)
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. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

9. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in Adam. arXiv preprint arXiv: 1711.05101 (2017)
10. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: a multitask benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
11. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142–150. Association for Computer Linguistics, Portland, Oregon, United States (2011)

Few-Shot Table-to-Text Generation with Structural Bias Attention

Di Liu[1,2,3], Weihua Wang[1,2,3](✉), Feilong Bao[1,2,3], and Guanglai Gaov[1,2,3]

[1] College of Computer Science, Inner Mongolia University, Hohhot, China
wangwh@imu.edu.cn

[2] National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot, China

[3] Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot, China

Abstract. Table-to-text generation task refers to converting tabular data into language text to facilitate easier understanding and analysis of the table. Recently, pre-trained models have made significant advancements in this kind of tasks. However, the inherent structural differences between tabular data and text, and the lack of domain-specific knowledge in few-shot datasets, make it challenging for pre-trained models to generate faithful text. To solve these problems, we proposed a framework that encodes tables by obtaining structural bias attention through pruning full self-attention, distinguishing the importance of cells from a structural perspective. We use the pre-trained model with the structural bias framework to the generation component of Prototype-to-Generation. To encourage prototype memory to adhere to the table content and generate more accurate and aligned sentences, we employ Reinforcement Learning. We conducted extensive experiments on three few-shot table datasets. Compared to previous advanced methods, our model achieved superior performance across multiple evaluation metrics.

Keywords: Few-shot Generation · Table-to-text Generation · Structural Bias Attention

1 Introduction

Table-to-text generation task aims to generate natural language descriptions for the key information within a table. It has found significant applications in various domains, including biography summarization [1], report generation [2], and question answering [3]. With the rapid development of neural networks, researchers such as Liu et al. [4], have proposed various neural models to address this problem. While these models have achieved promising results, they require a significant amount of table data for training. Due to the scarcity of table data, it has hindered the widespread use of neural models in current applications. With the emergence of pre-trained models, natural language generation task based on these models have demonstrated remarkable generation capabilities. However, the main challenge of applying pre-trained models to tabular datasets is the structural disparity between tables and text.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 349–354, 2024.
https://doi.org/10.1007/978-981-99-7022-3_31

In this work, we propose a structural bias framework (SF) to address the structural disparity between tables and text. Furthermore, to achieve better generation performance, we also applied the P2G [5]. It can alleviate the drawback of pre-trained models lacking domain-specific knowledge. Our framework (SF) can be integrated into pre-trained models, applying the enhanced language model to the generation component of the P2G. This allows for a more comprehensive capture of the structural information in tables and is particularly suitable for table-to-text generation under the few-shot scenarios. We conducted extensive experiments on three open-domain table datasets, and our method outperformed the state-of-the-art approaches across multiple evaluation metrics. In summary, our main contributions are as follows:

- We propose a structural bias framework and integrated it into a pre-trained model. Furthermore, we integrate the improved model into the P2G.
- We conduct extensive experiments and analysis on three open-domain table data sets, demonstrating the effectiveness of our method.

2 Methodology

2.1 Preliminaries

In our approach, the training dataset $D = \{(T, Y)_i\}_{i=1}^{|D|}$, where $T_i = t_1, \ldots, t_{|T|}$ represents linearized representations of tables. In addition, $Y_i = (y_1, \ldots, y_{|Y|})$ refers to the reference text, and y_i represents the textual description corresponding to table T_i.

2.2 Structural Bias Framework (SF)

Transformer [6] utilizes self-attention to capture information about all tokens in the input sequence. But this attention can't capture the key structural of the table. In tables, cells within the same row or column are semantically related. We consider cells that aren't in the same row or column as being unrelated to the table structure. Based on this feature, we propose a structural bias attention as shown in Fig. 1, which captures the structural dependencies in tables. Specifically, to extract the structural bias attention, we remove the attention that is unrelated to the table structure from the overall attention. We only retain the attention within the cells and the attention between related cells in the same row or column. This process allows us to focus on the important structural elements of the table and discard irrelevant attention connections.

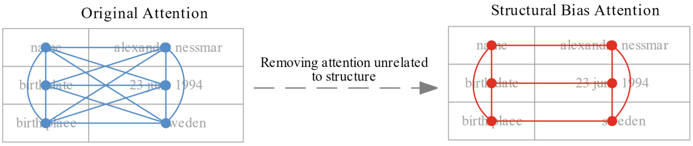
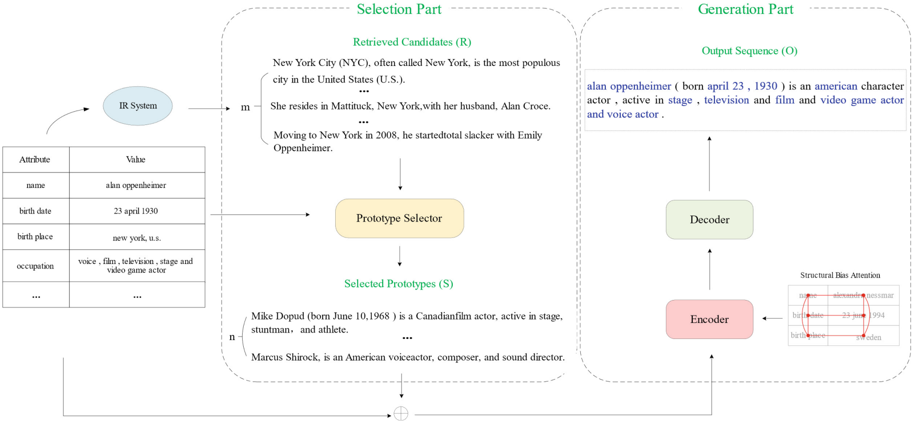


Fig. 1. An example of original attention and structural bias attention. In this example, we omit attention between tokens within the same cells.

2.3 Table-to-Text Generation Models

By incorporating the structural bias framework into the pre-trained model, we apply it to the generation component of the P2G. The overall architecture of our approach is illustrated in Fig. 2.



Attribute	Value
name	alan oppenheimer
birth date	23 april 1930
birth place	new york, u.s.
occupation	voice , film , television , stage and video game actor
...	...

New York City (NYC), often called New York, is the most populous city in the United States (U.S.).

...

She resides in Mattituck, New York,with her husband, Alan Croce.

...

Moving to New York in 2008, he startedtotal slacker with Emily Oppenheimer.

Mike Dopud (born June 10,1968) is a Canadianfilm actor, active in stage, stuntman， and athlete.

...

Marcus Shirock, is an American voiceactor, composer, and sound director.

alan oppenheimer (born april 23 , 1930) is an american character actor , active in stage , television and film and video game actor and voice actor .

Fig. 2. The overall architecture diagram of our method applied to the P2G generation part. The P2G is divided into two components: the prototype selector and the generator.

Structural Bias LR Training. For a given input data $D_i = \{(T, Y, S)\}$ and a sample output sequence $O = \{o_1, \ldots, o_{|O|}\}$, the RL training objective is defined as:

$$\mathcal{L}_{RL} = -R(Y, O) \sum_{i=1}^{|O|} \log P(Y_i | Y_{<i}, E(T, S)) \tag{1}$$

where $E(\cdot)$ represents the encoder module. S represents the prototype memory generated by the P2G model. The reward function $R(Y, O)$ represents the similarity between the generated text and the reference text. Define $R(Y, O) = B(Y, O)$, where $B(\cdot, \cdot)$ represents the BLEU score [7].

Learning Objective. We divide the learning process of the model into two stages. In the first stage, we employ traditional conditional language modeling learning objective:

$$\mathcal{L}_{LM} = -\sum_{i=1}^{|Y|} \log P(Y_i | Y_{1:i-1}, E(T, F)) \tag{2}$$

where F represents the key structural information from the table. In the second stage, the learning objective is shown in Eq. (3).

$$\mathcal{L}_{mix} = \mathcal{L}_{RL} + \mathcal{L}_{LM} \tag{3}$$

3 Experiments and Results

3.1 Datasets and Baselines

We evaluate our approach on three popular few-shot datasets [8]. We consider advanced few-shot table-to-text generation methods as baselines, including Switch + PLM [8], TableGPT [9], Prefix-Tuning [10], AMG [11], PCG [12], and P2G [5]. We chose T5[13] as the base pre-trained language model for our experiments.

3.2 Results

We selected BLEU-4 [7] and utilized the F1 score of PARENT [14], denoted as PARENT-F. Additionally, the best performance is indicated in bold, and the second best is marked with an underline. All (R) values are from the original paper (Table 1 and 2).

Table 1. The BLEU-4 results for the three datasets.

Domain	Humans	Books	Songs
Training set size	50 100 200 500	50 100 200 500	50 100 200 500
Switch + GPT-2(R)	25.7 29.5 36.1 41.7	34.3 36.2 37.9 40.3	36.1 37.2 39.4 42.2
TableGPT(R)	29.8 34.5 40.6 45.6	35.1 37.3 38.5 41.6	36.7 37.8 39.3 42.3
Prefix-Tuning + T5(R)	34.5 39.9 41.6 44.1	35.5 37.3 39.6 41.2	37.5 38.5 40.0 41.1
AMG(R)	- - - 49.0	- - - 43.9	- - - 45.1
PCG(R)	39.9 43.3 45.8 49.4	36.6 36.9 39.0 45.6	38.0 41.7 42.5 44.5
P2G (R)	39.3 42.6 46.2 50.1	41.2 43.4 46.4 49.2	42.8 45.9 47.6 50.7
Ours	**41.7 44.3 46.5 50.2**	**42.4 44.2 47.5 49.7**	**49.3 50.2 51.9 52.6**

Table 2. Here are the PARENT-F results for the three datasets. Baseline models that weren't evaluated with PARENT aren't shown in this table.

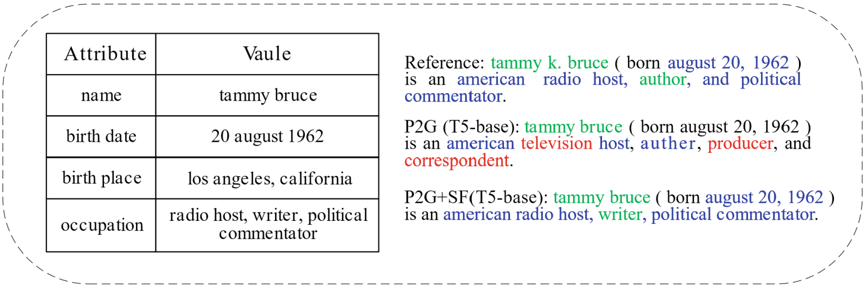
Domain	Humans	Books	Songs
Training set size	50 100 200 500	50 100 200 500	50 100 200 500
Switch + GPT-2(R)	30.6 34.6 40.5 45.6	42.7 42.8 43.4 44.9	40.2 41.7 44.0 44.8
Prefix-Tuning + T5(R)	39.3 40.6 41.8 42.1	32.8 34.8 36.0 36.8	34.4 36.1 36.0 34.6
AMG(R)	43.6 47.7 50.1 **51.9**	43.4 46.0 **47.5** 48.6	42.0 43.3 45.9 46.9
PCG(R)	**46.7 48.3 50.4** 51.8	**46.3** 46.2 **47.5 49.3**	**44.8 45.7 46.9** 46.0
Ours	46.2 47.9 50.2 51.7	46.1 **46.4** 47.4 49.1	44.1 45.2 46.5 **47.1**

Our method outperformed other baselines in terms of BLUE scores, demonstrating that our approach has the best overall performance. From the results, it can be observed that P2G achieved the second-best performance. We choose to apply our method within the P2G because it contributes to the improvement of the model.

For fidelity-based PARENT scores, although our method does not achieve the highest scores in all categories, the overall performance is considered second-best. This is because the PCG module utilizes BART-large as the base model for the generation part, which has stronger text generation capabilities. However, the average PARENT score difference between our model and theirs is around 0.4.

3.3 Case Study

We will utilize the case presented in Fig. 3 to visually demonstrate the effectiveness of our approach. It can be observed from Fig. 3 that our approach generates information that encompasses all the content of the table. On the other hand, when using P2G alone for generation, it includes partial table content and generates erroneous information that deviates significantly from both the table content and the reference text. This demonstrates that our framework can reduce the generation of erroneous information by P2G.



Attribute	Vaule
name	tammy bruce
birth date	20 august 1962
birth place	los angeles, california
occupation	radio host, writer, political commentator

Reference: tammy k. bruce (born august 20, 1962) is an american radio host, author, and political commentator.

P2G (T5-base): tammy bruce (born august 20, 1962) is an american television host, auther, producer, and correspondent.

P2G+SF(T5-base): tammy bruce (born august 20, 1962) is an american radio host, writer, political commentator.

Fig. 3. Comparison between the generation results of P2G based on our method and using P2G alone. Blue represents correct text, red represents incorrect text, and green represents text that aligns with the table but is not completely consistent with the reference text.

4 Conclusion

In this paper, we propose a structural bias framework that effectively reduces the generation of text by the P2G that don't align with the table content. The framework captures the crucial information from the table using structural bias attention, which guides the model in generating aligned text. We conduct experiments on three datasets to demonstrate the effectiveness of our approach.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Nos.62066033, 61966025); Inner Mongolia Key Research and Development Fund Project (No. 2023YFSW0001); Inner Mongolia Autonomous Region Over-seas Students Innovation and Entrepreneurship Startup Program.

References

1. Lebret, R., Grangier, D., Auli, M.: Neural text generation from structured data with application to the biography domain. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2016)
2. Hasan, S.A., Farri, O.: Clinical natural language processing with deep learning. Data Sci. Healthcare: Methodol. Appli., 147–171 (2019)
3. Li, Y., Li, W., Nie, L.: MMCoQA: conversational question answering over text, tables, and images. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4220–4231 (2022)
4. Liu, T., Wang, K., Sha, L., et al.: Table-to-text generation by structure-aware seq2seq learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32(1) (2018)
5. Su, Y., Meng, Z., Baker, S., et al.: Few-shot table-to-text generation with prototype memory. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 910–917 (2021)
6. Vaswani, A., Shazeer, N,, Parmar, N., et al.: Attention is all you need. Adv. Neural Inform. Proc. Syst. **30** (2017)
7. 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)
8. Chen Z, Eavani H, Chen W, et al. Few-Shot NLG with Pre-trained language model[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 183–190
9. Gong, H., Sun, Y., Feng, X., et al.: Tablegpt: few-shot table-to-text generation with table structure reconstruction and content matching. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1978–1988 (2020)
10. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. 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. 4582–4597 (2021)
11. Zhao, W., Liu, Y., Wan, Y., Yu, P.: Attend, memorize and generate: towards faithful table-to-text generation in few shots. In: Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, pp. 4106–4117. Association for Computational Linguistics (2021)
12. Luo, Y., Lu, M., Liu, G., Wang, S.: Few-shot table-to-text generation with prefix-controlled generator. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 6493–6504 (2022)
13. 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)
14. Dhingra, B., Faruqui, M., Parikh, A., Chang, M.-W., Das, D., Cohen, W.: Handling divergent reference texts when evaluating table-to-text generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4884–4895 (2019)

Generalized Knowledge Distillation for Topic Models

Kohei Watanabe(✉) and Koji Eguchi(✉)

Graduate School of Advanced Science and Engineering, Hiroshima University,
Higashi-Hiroshima, Japan
{m224406,kxeguchi}@hiroshima-u.ac.jp

Abstract. Topic modeling is used in the analysis of textual data to estimate the underlying topics within the dataset. Knowledge distillation has been attracting attention as a means of transferring knowledge from a large teacher model to a small student model in the field of deep learning. Knowledge distillation can be categorized into three types depending on the type of knowledge to be distilled: response-based, feature-based, and relation-based. To the best of our knowledge, previous studies on knowledge distillation used in topic models have all focused on response and/or feature knowledge, but these methods cannot transfer the structural knowledge of the teacher model to the student model. To solve this problem, we propose a generalized knowledge-distillation method that combines all three types of knowledge distillation, including the relation-based knowledge distillation with contrastive learning, which had not been used for neural topic models. Our experiments show that our neural topic model, trained with the proposed method, improves topic coherence compared to baseline models without knowledge distillation.

Keywords: Topic models · Knowledge distillation · Contrastive learning

1 Introduction

Topic modeling is a common method for estimating latent topics behind data from documents and has been applied to various tasks. A typical topic model, latent Dirichlet allocation (LDA) [2], generates documents probabilistically assuming that there are multiple latent topics behind each document. LDA is typically trained using variational Bayesian methods; however, the challenge is that a new inference process needs to be mathematically derived depending on the purpose of the model. Neural topic models have been proposed to solve this problem. One such model is Srivastava et al.'s PRODLDA [8], which is based on a variational autoencoder (VAE) [6]. It can approximate complex posterior distributions using a flexible inference network that is based on neural networks.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 355–361, 2024.
https://doi.org/10.1007/978-981-99-7022-3_32

In deep learning, knowledge distillation has attracted attention as a method for transferring knowledge from a large-scale teacher model to a small-scale student model. Knowledge distillation can be classified into three types depending on the type of knowledge to be distilled: response-based, feature-based, and relation-based [4]. In a previous study on knowledge distillation for neural topic models, Hoyle et al. proposed a response-based knowledge-distillation method that trains student neural topic models using the output of BERT, which is pretrained on large corpora, as the teacher model [5]. Adhya et al. also conducted response-based and feature-based knowledge distillation simultaneously using a large neural topic model as the teacher and a small neural topic model as the student [1]. However, these methods focus only on the individual sample representations, which means that they are unable to transfer structural knowledge, the relationships between samples, from the teacher model to the student model.

To solve this problem, we propose a relation-based knowledge-distillation method using contrastive learning for neural topic models. The method uses contrastive loss to distill the structural knowledge of the teacher by learning the latent representations of the student model, while maintaining the relationships in the individual document representations generated by the teacher model. We further propose a generalized knowledge distillation by combining response-based, feature-based, and relation-based knowledge distillation. Through evaluation experiments measuring topic coherence, we show that the neural topic model trained using the proposed method improves on a baseline neural topic model [3] and its variant.

2 Overview of Neural Topic Models

As an earlier neural topic model, PRODLDA [8] was developed using VAE [6]. A generalization of PRODLDA is SCHOLAR [3]. These neural topic models replace the Dirichlet prior used in the original LDA [2] with a logistic normal prior ($\mathcal{LN}$) to facilitate inference. Now suppose $\boldsymbol{w}_i^{\mathrm{BoW}}$ is a V-dimensional vector counting the words in document $\boldsymbol{w}_i$, and $\boldsymbol{z}_i$ is its corresponding topic vector. The VAE-based neural topic model learns to minimize the Kullback-Leibler (KL) divergence between the true posterior distribution $p(\boldsymbol{z}_i)$ and variational distribution $q(\boldsymbol{z}_i|\boldsymbol{w}^{\mathrm{BoW}})$, which cannot be obtained analytically. The evidence lower bound (ELBO) is expressed as

$$\mathrm{ELBO} = \mathbb{E}_{q(\boldsymbol{z}_i|\cdot)}\left[\mathcal{L}_{RE}\right] - \mathrm{D}_{\mathrm{KL}}\left[q\left(\boldsymbol{z}_i \mid \boldsymbol{w}_i^{\mathrm{BoW}}\right) \| p\left(\boldsymbol{z}_i \mid \alpha\right)\right], \tag{1}$$

where $\mathcal{L}_{RE} = (\boldsymbol{w}_i^{\mathrm{BoW}})^{\top} \log \sigma(\boldsymbol{\eta}_i)$. The notation $\sigma(\cdot)$ is a softmax function, $\sigma(\boldsymbol{\eta}_i)$ corresponds to the word distribution (multinomial distribution over the vocabulary) of document $\boldsymbol{w}_i$, $\mathcal{L}_{RE}$ is the reconstruction error, and $\mathrm{D}_{\mathrm{KL}}\left[q\left(\boldsymbol{z}_i \mid \boldsymbol{w}_i^{\mathrm{BoW}}\right) \| p\left(\boldsymbol{z}_i \mid \alpha\right)\right]$ is the KL divergence between $q(\boldsymbol{z}_i|\boldsymbol{w}_i^{\mathrm{BoW}})$ and $p(\boldsymbol{z}_i|\alpha)$. As in VAE, the inference process uses a multilayer neural network to generate the variational parameters. Since the logistic normal distribution is assumed for the prior distribution of $\boldsymbol{z}$, the inference network outputs a mean vector $\boldsymbol{\mu}(\cdot)$ and diagonal covariance matrix $\boldsymbol{\sigma}^2(\cdot)$. The variational distribution is $q(\boldsymbol{z}_i \mid \boldsymbol{w}_i^{\mathrm{BoW}}) = \mathcal{LN}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$.

$$\boldsymbol{\mu}_i = \mathbf{W}_\mu \boldsymbol{\pi}_i + \boldsymbol{b}_\mu, \quad \log \boldsymbol{\sigma}_i^2 = \mathbf{W}_\sigma \boldsymbol{\pi}_i + \boldsymbol{b}_\sigma, \quad \boldsymbol{\pi}_i = f\left(\mathbf{W}_w \boldsymbol{w}_i^{\mathrm{BoW}}\right), \tag{2}$$

where f is the multilayer perceptron and the variational parameters are all the weight matrices $\mathbf{W}_w$, $\mathbf{W}_\mu$, and $\mathbf{W}_\sigma$ and biases $\boldsymbol{b}_\mu$ and $\boldsymbol{b}_\sigma$ in Eq. (2).

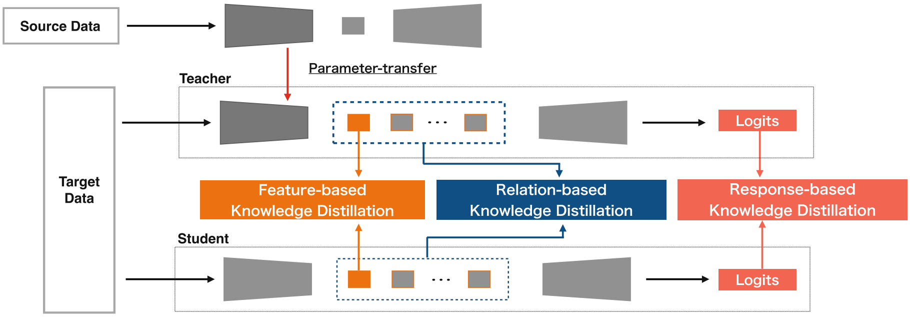

Source Data
Parameter-transfer
Teacher
Target Data
Logits
Feature-based Knowledge Distillation
Relation-based Knowledge Distillation
Response-based Knowledge Distillation
Student
Logits

Fig. 1. Conceptual diagram of generalized knowledge distillation.

3 Methodology

On the basis of the neural topic model SCHOLAR [3], our method unify response-based and feature-based knowledge distillation using transfer learning and relation-based knowledge distillation using contrastive learning. It differs from previous methods in that we apply relation-based knowledge distillation [9] to the neural topic model, which has not been studied previously, and in that we propose to integrate the three types of knowledge distillation in a unified framework. As knowledge distillation require s employing an identical dataset for both student and teacher models, we initialize the teacher model's weight matrix $\mathbf{W}_w$ for the target data by leveraging the weight matrix $\mathbf{W}_w$ pre-trained on a source data. Figure 1 shows a conceptual diagram of generalized knowledge distillation.

For the inference process of neural topic models described in Sect. 2, we use the following objective function instead of $\mathcal{L}_{RE}$ in Eq. (1),

$$\mathcal{L} = (1-\gamma)\mathcal{L}_{RE} + \gamma\mathcal{L}_{ResKD} + \lambda_1\mathcal{L}_{FeaKD} + \lambda_2\mathcal{L}_{RCD}. \tag{3}$$

Here, $\mathcal{L}_{ResKD}$, $\mathcal{L}_{FeaKD}$, and $\mathcal{L}_{RCD}$ corresponds to response-based, feature-based, and relation contrastive distillation, respectively. The details of these terms are explained in the rest of this section. The notations $\gamma, \lambda_1, \lambda_2$ are hyper-parameters to adjust the effect of each term.

Response-Based Knowledge Distillation: The generative process of the models trained with our proposed method is the same as that of SCHOLAR. The inference process uses the SCHOLAR inference network but adds a pseudo-document $\boldsymbol{w}_i^t$ to Eq. (2), which is generated from the logit of the teacher model.

$$\boldsymbol{\pi}_i = f\left(\left[\mathbf{W}_w \boldsymbol{w}_i^{\mathrm{BoW}}; \mathbf{W}_{w^t} \boldsymbol{w}_i^t\right]\right), \tag{4}$$

where $\left[\mathbf{W}_w \boldsymbol{w}_i^{\mathrm{BoW}}; \mathbf{W}_{w^t} \boldsymbol{w}_i^t\right]$ denotes the horizontal concatenation of $\mathbf{W}_w \boldsymbol{w}_i^{\mathrm{BoW}}$ and $\mathbf{W}_{w^t} \boldsymbol{w}_i^t$. To apply knowledge distillation to a neural topic model, the following objective function $\mathcal{L}_{ResKD}$ is used

$$\mathcal{L}_{ResKD} = \tau^2 (\boldsymbol{w}_i^t)^\top \log \hat{\boldsymbol{w}}_i, \quad \boldsymbol{w}_i^t = \sigma(\boldsymbol{\eta}_i^t/\tau) N_i, \quad \hat{\boldsymbol{w}}_i = \sigma(\boldsymbol{\eta}_i/\tau), \tag{5}$$

where $\boldsymbol{w}_i^t$ is the probability estimated from the logit $\boldsymbol{\eta}_i^t$ of the teacher model, scaled by the document length N and treated as a smoothed pseudo-document, and τ is the temperature of the softmax function.

Feature-Based Knowledge Distillation: Feature-based knowledge distillation distills the topic multinomial distribution of the documents from the teacher model to the student model as knowledge. The objective function of feature-based knowledge distillation is expressed as

$$\mathcal{L}_{FeaKD} = -\sum (\boldsymbol{z}_i^t - \boldsymbol{z}_i^s)^2 \tag{6}$$

where $\boldsymbol{z}_i^t$ and $\boldsymbol{z}_i^s$ indicate the latent representations (i.e., features or topics) generated by the teacher and student models, respectively, for document $\boldsymbol{w}_i$.

Relation Contrastive Distillation: Now, we describe the method for achieving relation-based knowledge distillation by maximizing the mutual information of the relation Y^t between the latent representations of the teacher model and that $Y^{t,s}$ between the latent representations of the teacher model and student model. The idea is inspired by [9]; however, we employ it in the context of inference of neural topic models. Let $p(W)$ be the empirical distribution of the document set $W = \{\boldsymbol{w}_i : i = 1, ..., D\}$ of the training data and model the conditional marginal distributions of topic relations $p(Y^t|W)$ and $p(Y^{t,s}|W)$ as follows.

$$\boldsymbol{w}_i, \boldsymbol{w}_j, \boldsymbol{w}_m, \boldsymbol{w}_n \sim p(W), \quad \boldsymbol{y}_{i,j}^t = g^t(\boldsymbol{z}_i^t, \boldsymbol{z}_j^t), \quad \boldsymbol{y}_{m,n}^{t,s} = g^{t,s}(\boldsymbol{z}_m^t, \boldsymbol{z}_n^s), \tag{7}$$

where $\boldsymbol{z}_i^t$ is the latent representation generated by the decoder of the teacher neural topic model for document $\boldsymbol{w}_i$, and $\boldsymbol{z}_n^s$ is that generated by the student neural topic model for document $\boldsymbol{w}_n$. The g^t is a network that computes the relation between the latent representations of the teacher model and $g^{t,s}$ is a network that computes the relation between the latent representations of the teacher model and student model. We also model $p(Y^t, Y^{t,s}|W)$ as follows.

$$\boldsymbol{w}_i, \boldsymbol{w}_j \sim p(W), \quad \boldsymbol{y}_{i,j}^t = g^t(\boldsymbol{z}_i^t, \boldsymbol{z}_j^t), \quad \boldsymbol{y}_{i,j}^{t,s} = g^{t,s}(\boldsymbol{z}_i^t, \boldsymbol{z}_j^s). \tag{8}$$

The mutual information of $p(Y^t|W)$ and $p(Y^{t,s}|W)$ is expressed as follows.

$$I(Y^t, Y^{t,s}) = \mathbb{E}_{p(Y^t,Y^{t,s}|W)} \log \frac{p(Y^t, Y^{t,s}|W)}{p(Y^t|W)p(Y^{t,s}|W)}. \tag{9}$$

To derive the objective function, we define a latent variable δ that indicates whether the relation pairs $(\boldsymbol{y}^t, \boldsymbol{y}^{t,s})$ are generated from the joint distribution or product of marginal distributions. When $\delta = 1$, it means that $\boldsymbol{y}^t$ and $\boldsymbol{y}^{t,s}$ are computed by the same input pair, as in Eq. (8), and when $\delta = 0$, it means that $\boldsymbol{y}^t$ and $\boldsymbol{y}^{t,s}$ are computed by independently selected input pairs, as in Eq. (7). Maximizing the mutual information is equivalent to maximizing the following objective function $\mathcal{L}_{RCD}$ of relation contrastive distillation [9].

$$\mathcal{L}_{RCD} = \sum_{q(\delta=1)} \log h(\boldsymbol{y}^t, \boldsymbol{y}^{t,s}) + N \sum_{q(\delta=0)} \log[1 - h(\boldsymbol{y}^t, \boldsymbol{y}^{t,s})], \tag{10}$$

where$\{(\boldsymbol{y}^t, \boldsymbol{y}^{t,s})|\delta = 1\}$ is a positive pair and $\{(\boldsymbol{y}^t, \boldsymbol{y}^{t,s})|\delta = 0\}$ is a negative pair, and N is the number of negative pairs for a positive pair. h is a model for approximating true distribution $q(\delta = 1|Y^t, Y^{t,s})$, where $h : \{Y^t, Y^{t,s}\} \to [0, 1]$. Not only h, but also the student network and subnetworks are optimized when $\mathcal{L}_{RCD}$ is minimized.

Table 1. Datasets that differ in total number of documents D and vocabulary size. V

	Wiki (Source)	IMDb (Target)	20NG (Target)	BBC (Target)
D	6,078,287	50,000	18,745	2,225
V	50,000	5,000	1,995	9,635

Table 2. NPMI and sample standard deviation.

Model	IMDb	20NG	BBC
SCHOLAR	0.164 (0.006)	0.316 (0.005)	0.279 (0.011)
SCH.+Wiki	0.162 (0.003)	0.321 (0.003)	0.280 (0.006)
SCH.+ResKD+FeaKD+RCD	**0.167 (0.002)**	**0.349 (0.010)**	**0.321 (0.012)**

4 Experiments and Results

We used the English Wikipedia dataset (Wiki)[1] as the source data for pretraining SCHOLAR, and the IMDb dataset of movie reviews (IMDb)[2], 20News-

[1] https://huggingface.co/datasets/wikipedia.

[2] http://ai.stanford.edu/~amaas/data/sentiment/.

groups dataset (20NG)[3], and BBC dataset (BBC)[4] as the target data to be analyzed. We split the datasets into training, development, and test sets (train/dev/test) in the following proportions: 20NG: 48/12/40, IMDb: 50/25/25, BBC: 70/15/15. The vocabulary of the Wiki dataset used for the pre-training was formed by keeping the top 50,000 words that occurred in most documents. Details of the datasets are listed in Table 1. We set the number of topics to 50 in the evaluation experiment. We used Optuna[5] to tune the hyperparameters τ, γ, λ_1, and λ_2.

The models trained with the proposed method were evaluated using normalized pointwise mutual information (NPMI) [7], a measure of topic coherence based on the co-occurrence of words in a corpus, using a test set of the top 10 words for each topic in the same corpus. Table 2 lists the experimental results. The NPMI in the table is the average of five runs with different random initialization. The baseline models are SCHOLAR [3] and SCH.+Wiki, which was trained by transferring parameters from the SCHOLAR pre-trained on the large dataset, i.e., Wiki, and used as a teacher model in the knowledge distillation. The model (SCH.+ResKD+FeaKD+RCD) trained using the proposed method, which combines the three types of knowledge distillation (response-based, feature-based and relation-based), achieved the best NPMI on all three datasets compared with the two baselines: SCHOLAR [3] and SCH.+Wiki. We found that the SCH.+Wiki achieved better NPMI than the original SCHOLAR on the 20NG and BBC datasets, but slightly worse on the IMDb dataset.

5 Conclusions

We proposed a generalized knowledge distillation for training neural topic models, by unifying three types of knowledge distillation: response-based, feature-based, and relation-based. The response-based and feature-based knowledge-distillation are based on parameter transfer from a teacher model trained with a larger dataset. The relation-based knowledge distillation is based on contrastive learning that transfers topic relationships of a teacher model into a student model. This is the first work on relation-based knowledge distillation for neural topic models, to our knowledge. Evaluation experiments indicated that all three types of knowledge distillation improved the performance of the neural topic models trained with our method in several datasets. For future work, we plan to investigate which type of teacher is best suited for each of dataset to be analyzed. The use of large language models as teacher models is also a possible extension of our work.

Acknowledgements. This work was supported in part by the Grant-in-Aid for Scientific Research (#23K11231) from JSPS, Japan, and in part by ROIS NII Open Collaborative Research 2023 (#23FS02).

[3] https://github.com/akashgit/autoencoding_vi_for_topic_models.

[4] http://mlg.ucd.ie/datasets/bbc.html.

[5] https://optuna.org/.

References

1. Adhya, S., Sanyal, D.K.: Improving neural topic models with Wasserstein knowledge distillation. In: Kamps, J., et al. (eds.) ECIR 2023. LNCS, vol. 13981, pp. 321–330. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28238-6_21
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. JMLR **3**, 993–1022 (2003)
3. Card, D., Tan, C., Smith, N.A.: Neural models for documents with metadata. In: ACL 2018, pp. 2031–2040 (2018)
4. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. IJCV **129**, 1789–1819 (2021)
5. Hoyle, A.M., Goel, P., Resnik, P.: Improving neural topic models using knowledge distillation. In: EMNLP 2020, pp. 1752–1771 (2020)
6. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: ICLR 2014 (2014)
7. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: EACL 2014, pp. 530–539 (2014)
8. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. In: ICLR 2017 (2017)
9. Zhu, J., et al.: Complementary relation contrastive distillation. In: CVPR 2021, pp. 9260–9269 (2021)

Improving Speaker Recognition by Time-Frequency Domain Feature Enhanced Method

Jin Han[1], Yunfei Zi[1], and Shengwu Xiong[1,2](✉)

[1] College of Computer and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
xiongsw@whut.edu.cn

[2] Sanya Science and Education Innovation Park, Wuhan University of Technology, Wuhan 572000, Hainan, China

Abstract. Many existing speaker recognition algorithms have the problem that single-domain feature extraction cannot represent the speech characteristics well, and this problem will affect the accuracy of speaker recognition. To solve this problem, we propose a time-frequency domain feature enhanced deep speaker (TFDS). The proposed algorithm can combine time domain and frequency domain, enhance the traditional MFCC feature extraction, and make up for the shortcomings of other algorithms that only extract features in a single domain. The deep speaker network architecture includes ResCNN, GRU, time averaging layer, style transformation layer, length normalization layer, and the loss is triple loss. Representation of experimental results performed on the librisspeech dataset results show that TFDS has higher accuracy and lower Equal Error Rate than deep speaker, and the time-frequency domain feature enhanced method can also be combined with other networks to improve the accuracy of speaker recognition.

Keywords: speaker recognition · feature enhancement · time-frequency domain · deep speaker · deep learning

1 Introduction

Speaker recognition has been used in information security, financial payment, intelligent hardware, attendance authentication and fraud control [1,6]. Traditional methods to solve this problem are mainly based on GMM to map voice features into low-dimensional vectors [3].

In particular, [2] proposed a speaker recognition method for a new speaker embedding system, which maps the speech sentence to a hyperplane and then calculates the similarity between speakers by cosine similarity. The speaker recognition technology has been more mature, but the above papers have certain shortcomings, the key is that because the single domain feature extraction cannot be a good performance of speech features, there is no good feature vector

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 362–367, 2024.
https://doi.org/10.1007/978-981-99-7022-3_33

corresponding to speech. No amount of work on training and loss function is in vain, which will affect the accuracy of speaker recognition [5].

Therefore, this paper proposes an algorithm of time-frequency domain feature enhanced deep speaker (TFDS) for speaker recognition. Firstly, the algorithm extracts the speaker features through the feature enhancement method in time and frequency domain, and comprehensively considers the speech features of multiple domains of audio. Then, the deep speaker algorithm is used for speaker modeling and recognition to improve accuracy. The experimental results verify the effectiveness of the algorithm, and the time-frequency domain feature enhanced method can improve the accuracy.

2 Method

2.1 Time-Frequency Domain Feature Enhanced Method

In this part, we introduce the time-frequency domain feature enhanced method in detail, and its structure diagram is shown in Fig. 1. A piece of continuous speech has features in time domain and frequency domain. We add these two dimensions to enhance the feature vector extracted by MFCC from the perspective of multiple domains.

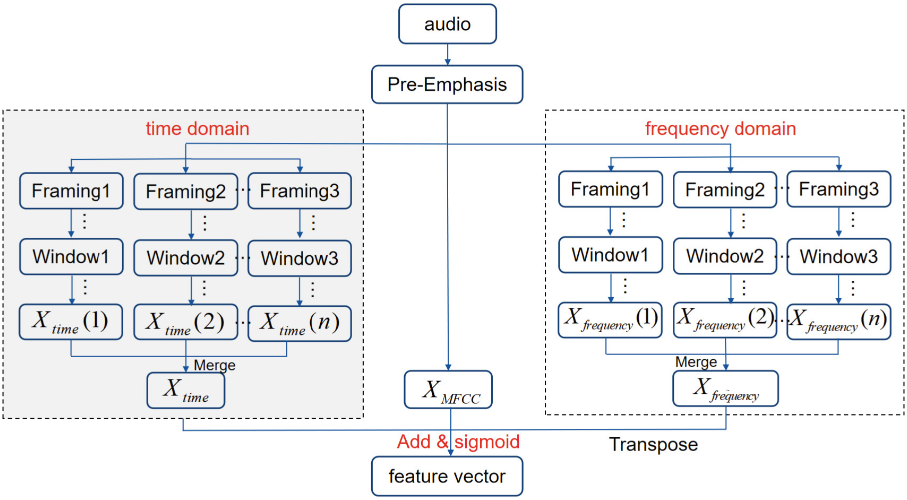


Fig. 1. Time-Frequency Domain Feature Enhanced Method

Time Domain Feature Enhanced Method. Firstly, a piece of continuous speech is pre-weighted. In the time domain, the pre-weighted continuous speech is segmented to obtain a speech segment within a short time.

The fast Fourier transform (FFT) is adopted. Through a Mel filter bank, the logarithmic energy of the output of each filter bank is calculated. After the

FFT transform, the convolution becomes a multiplication, and after taking the logarithm, the multiplication becomes an addition, converting the convolution signal into an additive signal. Finally, the MFCC coefficients of this small part of speech after framing are obtained by discrete cosine transform.

After the above calculation, the vector obtained after this framing is denoted as $X_{time}(1)$, Similarly, $X_{time}(2)$, ..., $X_{time}(n)$ can be obtained in the same way. Finally, the matrix merging operation is performed on $X_{time}(1)$, $X_{time}(2)$,...,$X_{time}(n)$ to obtain the feature vector based on time domain feature enhancement, which is denoted as X_{time}, and each of these elements is called $x_{time}(i)$.

Frequency Domain Feature Enhancement Considering that the relationship between the time domain and the frequency domain is the relationship between the horizontal axis and the vertical axis in the coordinate axis, the processing method for the frequency domain is mostly similar to the above processing for the time domain, except that the feature vector based on the feature enhancement in the frequency domain obtained at the end is denoted as $X_{frequency}$, and $X_{frequency}$ needs to transpose the matrix, and the transposed matrix is called $X^T_{frequency}(i)$. Each of these elements is called $x^T_{frequency}(i)$.

Time-Frequency Domain Feature Enhanced Method. In addition to the time domain and frequency domain calculation, this continuous speech also needs to go through a complete MFCC feature extraction process, and the obtained feature vector is denoted as X_{MFCC}, and each of these elements is called x_{MFCC}. The feature vector based on the time frequency domain enhanced method needs to add the above three parts element-by-element, which are the feature vector based on time domain enhancement, the feature vector based on time domain enhancement and the feature vector extracted by MFCC of the complete speech, and the formula is as follows:

$$x(i) = x_{MFCC}(i) + x_{time}(i) + x^T_{frequency}(i) \tag{1}$$

where the vector composed of all $x(i)$ is the enhanced feature vector in the time-frequency domain. Considering that there may be elements with values greater than 1 in this vector, a sigmoid activation function is needed to obtain the values in the (0,1) interval for the convenience of subsequent model training.

At this point, the feature vector based on the time-frequency domain enhanced method is computed.

2.2 Deep Speaker Network

In this subsection, we will introduce the deep speaker network, using the feature vectors extracted in Part 2.1 as the input of the network. The network architecture includes ResCNN, GRU, time averaging layer, stylized transformation layer, length normalization layer, and the loss is triplet loss. These architectures are further explained below.

3 Experimental Setup

In this subsection, we present the experimental setup of the time-frequency domain feature enhanced deep speaker (TFDS). The dataset we use is the train-clean-100 dataset from LibriSpeech, which contains about 100 h of clean speech.

The following experiments are carried out on the above data sets. They are GMM-UBM, x-vector/PLDA, i-vector/Cosine, i-vector/PLDA, x-vector/Softmax loss, d-vector, GAN, and the time-frequency domain feature enhanced deep speaker (TFDS). The Equal Error Rate corresponding to these algorithms under the data set is obtained as the evaluation index [4].

To prove that the feature enhancement method based on time-frequency domain can improve the accuracy of speaker recognition and reduce the Equal Error Rate, we set up the following four groups of ablation experiments. They are the method based on MFCC, the method based on time domain feature enhancement, the method based on frequency domain feature enhancement and the method based on time and frequency domain feature enhancement are used to extract features, and then input into the deep speaker network for training.

4 Results and Analysis

We will introduce the experimental results and result analysis of TFDS. The results on Equal Error Rates for GMM-UBM, x-vector/PLDA, i-vector/Cosine, i-vector/PLDA, GAN, and the time-frequency domain feature enhanced deep speaker (TFDS) are shown in Table 1. The results of the time-frequency domain feature enhanced method applied to deep speaker are shown in Fig. 2 and Fig. 3. And the results on accuracy and Equal Error Rate for four experiments are shown in Table 2.

Table 1. Performance comparison (EER) of the baseline model and the proposed method.

system	GMM-UBM	i-vector/Cosine	i-vector/PLDA	x-vector/PLDA
EER[%]	6.02	8.48	5.61	5.87
system	x-vector/Softmax loss	d-vector	GAN	TFDS
EER[%]	5.08	7.95	5.90	3.42

It can be found from Table 1 that GMM-UBM, i-vector/Cosine, i-vector/PLDA, x-vector/PLDA, x-vector/Softmax loss, d-vector, GAN have different effects on the task of speaker recognition in the case of the same data set and Equal Error Rate as the evaluation index. However, their effects are not as good as the effect of the time-frequency domain feature enhanced deep speaker (TFDS). The experimental results show that TFDS is more effective than other models mentioned in this paper under the same data set.

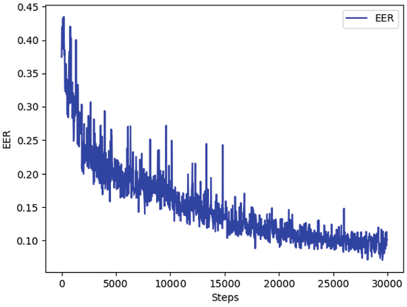


Fig. 2. Plot of Equal Error Rate as a function of number of iterations.

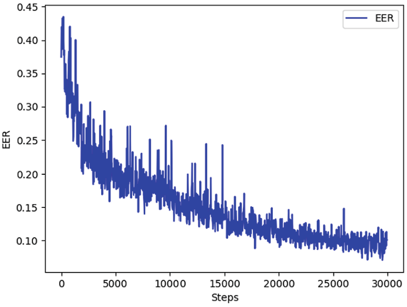


Fig. 3. Plot of Loss function, average loss function, accuracy, and average accuracy.

Figure 2 shows how the Equal Error Rate changes with the number of iterations. It can be found that the Equal Error Rate reaches the minimum value at step=24500, with a value of 3.42%.

The upper part of Fig. 3 shows the change of loss function and average loss function with the number of iterations, and the lower part shows the change of accuracy and average accuracy with the number of iterations. It can be found that the loss function is in a downward state as a whole and is becoming increasingly stable. With the increase of iterations, the accuracy is getting higher and closer to 100%. Its maximum value is 99.78%.

Table 2. Ablation experiment comparison of the proposed method (ACC, EER).

system	ACC[%]	EER[%]
$MFCC$/deep speaker	98.5714286	7.3686852
time domain feature enhanced deep speaker	99.1428571	7.1064142
frequency domain feature enhanced deep speaker	99.0327869	7.0801271
time-frequency domain feature enhanced deep speaker	99.7857143	3.4208765

We set up ablation experiments. It can be found from Table 2 that the time-frequency domain feature enhanced method is superior to the traditional MFCC method, the time-frequency domain feature enhanced method and the frequency domain feature enhanced method in terms of accuracy and EER. According to this result, we can conclude that, compared with single domain feature extraction, the time-frequency domain feature enhanced method can extract increasingly comprehensive speech feature information, which improves the accuracy and EER of voiceprint recognition to a certain extent. The time domain feature enhanced method and the frequency domain feature enhanced method has performance improvement compared with the traditional MFCC method in terms of accuracy. In addition, the time domain feature enhanced method is better than the frequency domain feature enhanced method in terms of accuracy, but it does not perform as well in terms of EER.

5 Conclusion

In this paper, we propose a TFDS algorithm for text-independent speaker recognition. This algorithm extracts features through time-frequency domain feature enhanced method, considers speech feature information in multiple domains, and then maps the features to a hyperplane through the deep speaker method, and trains the model through cosine similarity and triplet loss. The deep speaker network architecture includes ResCNN, GRU, time averaging layer, style transformation layer, length normalization layer.

Experiments show that under the same circumstances, the time-frequency domain feature enhanced deep speaker can achieve lower Equal Error Rate. Compared with the traditional GMM-UBM, x-vector/PLDA, x-vector/Softmax loss, d-vector, and GAN, TFDS algorithm can further improve the accuracy of speaker recognition based on deep speaker, which also shows that under the same circumstances, the feature enhancement based on time-frequency domain can improve the accuracy compared with the traditional MFCC. This method is hopeful to be applied to other networks to improve the accuracy of speaker recognition.

References

1. Chung, J.S., Nagrani, A., Zisserman, A.: VoxCeleb2: deep speaker recognition. Technical report (2018)
2. Cui, X., Goel, V., Saon, G.: Embedding-based speaker adaptive training of deep neural networks. arXiv preprint arXiv:1710.06937 (2017)
3. DiBiase, J., Silverman, H., Brandstein, M.: Microphone arrays: signal processing techniques and applications. In: chapter Robust Localization in Reverberant Rooms, pp. 157–180. Springer, Berlin (2001)
4. Nagrani, A., Chung, J.S., Zisserman, A.: VoxCeleb: a large-scale speaker identification dataset. In: Proceedings of the Interspeech 2017, pp. 2616–2620 (2017). https://doi.org/10.21437/Interspeech.2017-950
5. Pandey, A., Wang, D.: TCNN: temporal convolutional neural network for real-time speech enhancement in the time domain. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6875–6879. IEEE (2019)
6. Reynolds, D., et al.: The 2016 NIST speaker recognition evaluation (2017)

Leveraging Dual Encoder Models for Complex Question Answering over Knowledge Bases

Xin Wang[1], Honglian He[1](✉), Yongqing Diao[1], and Huayi Zhan[2](✉)

[1] Southwest Petroleum University, Chengdu, China
xinwang@swpu.edu.cn, 202121000475@stu.swpu.edu.cn,
202222000558@stu.swpu.edu.cn

[2] Sichuan Changhong Electric Co. Ltd., Mianyang, China
huayi.zhan@changhong.com

Abstract. Knowledge-based question answering is a hot topic in Natural Language Processing (NLP), especially in addressing complex questions. Existing methods, which transform complex questions into query graphs, often struggle with low-quality graphs. To improve this, we propose a dual-encoder model for generating and ranking query graphs. We incorporate beam search and a scoring function for high-quality graph generation, and use a dual-encoder model with attention mechanism for graph ranking. By extracting semantic structures from complex questions, we further refine the ranking process. Our experiments on benchmark datasets show competitive results, suggesting practical applications in complex question answering.

Keywords: Knowledge base · Question answering · Complex questions · Query graphs · Dual encoder model

1 Introduction

The main task of question answering over knowledge base (KBQA) is to find answers to questions from a knowledge base. The focus of research has shifted from simple QA to complex QA, which shows greater application potential. Existing methods mainly parse questions into query graphs and search for answers in the knowledge base. This process involves topic entity recognition and the challenging tasks of multi-hop path reasoning and constraint association.

To cope with aforementioned challenges, considerable prior work addresses core path reasoning and constraints association separately and suffers from poor performance due to excessive incorrect query graphs they produce. Recognizing this, we propose an approach to answering complex questions. In particular, our approach narrows down the search space for query graph generation through beam search and semantic structure matching, where dual encoders are incorporated. The experimental results on the benchmark dataset demonstrate the effectiveness of our proposed approach.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 368–374, 2024.
https://doi.org/10.1007/978-981-99-7022-3_34

Contributions. The contributions of the paper are summarized as follows.

- We introduce a module that applies the beam search along with a scoring function for query graph generation. Our scoring function accurately measures the quality of core paths of a question, by using the dual-encoder architecture.
- We develop a module to rank query graphs using a dual-encoder model. Moreover, we extract five typical semantic structures from complex questions and develop a classifier to predict the semantic structure of a question. The predicted semantic structure is used to filter out query graphs that do not conform to it.
- We conduct experimental studies on typical benchmark datasets. The results demonstrate promising improvements over existing techniques, highlighting the effectiveness of the proposed method in addressing complex question answering on knowledge bases.

2 Related Work

We categorize prior works into two types: Semantic Parsing (SP-based) [3,11,13] and Information Retrieval (IR-based) [1,2,5,10,12].

SP-Based. Semantic Parser typically converts a question posed in natural language into a logical structure, which can then be processed to procure the answer. [13] revisited the value of semantic parsing annotations and demonstrated that training models with annotated semantic parsing can significantly improve the performance of KBQA on a large-scale dataset. However, annotating semantic parsing is an expensive and time-consuming process. [3] offered a neural semantic parsing approach tailored for KBQA. It includes a retriever for efficiently retrieving relevant KB items, a transducer for generating logical forms with guaranteed grammatical correctness, and a checker for improving the transduction process. [11] improved question answering using comparative learning and transformers for entity linking and relationship prediction.

IR-Based. This technique generates candidate answers from a knowledge base according to the question's topic entity, then uses scoring methods to select the best answer based on information from the question and candidates. [1] introduced a novel approach to KBQA, where multi-constraint questions are converted into multi-constraint query graphs to address their complexities. [5] in knowledge-based relation extraction, the number of hops is not limited, thereby reducing the search space. However, the use of greedy search for each hop results in slower retrieval efficiency. [2] develop a methodology for answering complex queries, involving the use of simple queries to construct complex ones. [12] propose a sequential reasoning self-attention mechanism to address multi-hop reasoning. Lastly, [10] use an attention mechanism and a memory-based network to generate query graphs.

3 An Approach

Our approach to complex question answering uses a two-module system to find answers from a knowledge base. The modules, detailed in Sects. 3.1 and 3.2, generate and rank query graphs. See Fig. 1.

3.1 Query Graph Generation

Given a question q, the **Q**uery **G**raph **G**eneration module (QGG) produces a set of candidate query graphs, as follows.

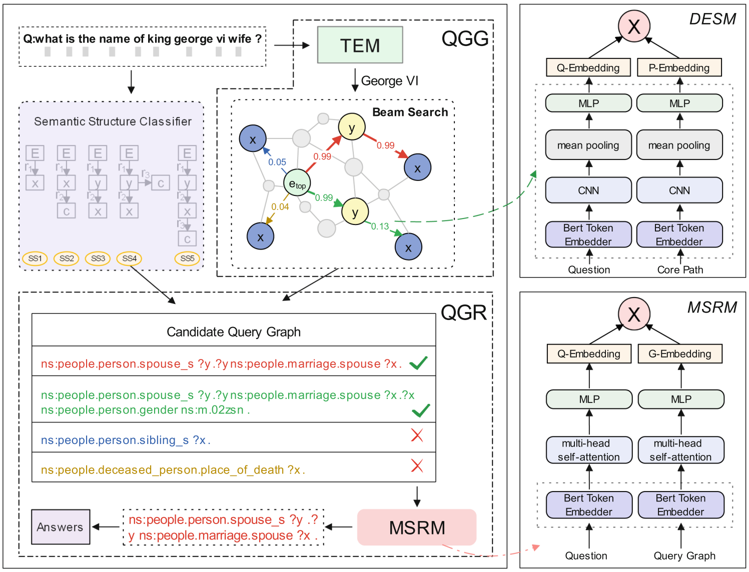

Q:what is the name of king george vi wife ?
TEM
QGG
George VI
Semantic Structure Classifier
E
r1
x
y
c
r2
r3
SS1
SS2
SS3
SS4
SS5
Beam Search
x
y
e_top
0.05
0.99
0.99
0.04
0.99
0.13
QGR
Candidate Query Graph
ns:people.person.spouse_s ?y .?y ns:people.marriage.spouse ?x .
ns:people.person.spouse_s ?y .?y ns:people.marriage.spouse ?x .?x ns:people.person.gender ns:m.02zsn .
ns:people.person.sibling_s ?x .
ns:people.deceased_person.place_of_death ?x .
✓
✓
✗
✗
MSRM
ns:people.person.spouse_s ?y .?y ns:people.marriage.spouse ?x .
Answers
DESM
X
Q-Embedding
P-Embedding
MLP
mean pooling
CNN
Bert Token Embedder
Question
Core Path
MSRM
X
Q-Embedding
G-Embedding
MLP
multi-head self-attention
Bert Token Embedder
Question
Query Graph

Fig. 1. Influence by varying ξ and η

Core Path Generation. QGG applies the traditional model BERT, and utilizes BiLSTM and CRF for span detection tasks. Entity linking is achieved via the Google Knowledge Graph API[1], linking entity mentions to candidates in a knowledge base. Once we have linked entities, we obtain the topic entity. Starting from the topic entity, the beam search is applied to generate core paths; meanwhile, QGG incorporates a scoring function to guide the search on KG, thereby reducing search cost and improving the quality of core paths. Specifically, the scoring function leverages a **D**ual **E**ncoder **S**coring **M**odel (DESM), whose architecture is shown in Fig. 1. For the model training, both the question q and the core path p_i pass through two towers, respectively to obtain two fixed-length feature

[1] https://developers.google.com/knowledge-graph.

vectors. The cosine similarity $cos(q, p_i)$ between two feature vectors is then calculated as the similarity of q and p_i. Training samples consist of positive samples and negative samples. Starting from the topic entity of a question, the one-hop path to the golden answer is picked as the positive sample, while the others are chosen as negative samples; starting from the one-hop positive sample of a question, the two-hop path to the golden answer is also considered as the positive sample while the others are marked as negative samples. A well trained DESM guides the search as follows. During the traversal, DESM calculates the weight of each hop and only k hops with the largest weights are considered to generate core paths. Here, parameter k is the beam size and used to restrict the number of branches.

Constraints Association. Another difficulty in complex question answering comes from various constraints taken by questions. Typically, constraints in complex questions are categorized as entity, type, time, and order constraints [9]. QGG associates each of them either to the entity node that is in the middle of a core path or the answer node of a core path. When dealing with an *entity constraint*, QGG simply determines whether the expanded node has a name in the question and treats it as an *entity constraint* if so. For *time constraints*, QGG recognizes them through regular matching. For *type constraints* and *order constraints*, manual rules are used to distinguish them.

3.2 Query Graphs Ranking

Given a set of candidate query graph sequences, we build a **Q**uery **G**raph **R**anking module (QGR) to score candidate query graphs. Via ranking, the query graph with the highest score is used to find the object in the KB as the final answer to the input question q. The core part of QGR is the **M**ulti-head **S**elf-attention **R**anking **M**odel (MSRM). We next introduce MSRM in details.

Model Details. Similar to DESM, MSRM also utilizes the dual-encoder architecture to compute the relevance between a question q and its query graphs $g_i^q (i = 1, 2, ...)$. As shown in Fig. 1, MSRM is a Siamese network with two identical towers, in which the parameters of the BERT layer are shared. For the model training, the question q and its query graph g_i^q pass through two towers, respectively to obtain two fixed-length feature vectors. The relevance between q and g_i^q is measured through cosine similarity between features of q and g_i^q. We use the MSEloss function to measure the overall similarity between question and query graph pairs.

Sample Strategy. For model learning, we applied the following sampling strategy to construct the training data for MSRM. For a given question q and its candidate query graphs, if a query graph can correctly (resp. incorrectly) find the answer of q, it is deemed as a positive (resp. negative) sample and assigned a value of 1 (resp. 0); otherwise, if a query graph can partially identify correct answers, *i.e.,* its F1 score ranges from 0 to 1, it is given a value $\eta \in (0, 1)$. Intuitively, the value quantifies the match degree of a pair of question and its candidate query graph, and serves as supervision for defining the loss function.

3.3 Optimization

To improve performance, we refine the candidate set of query graphs using a classification model that predicts semantic structures of questions, inspired by [8]. This model uses pre-trained BERT for question representation and is trained on data where the labels are semantic structures extracted from SPARQL queries of the questions. The optimization process refines the candidates by extracting a subset S_h from the candidate query graph set S_c. This subset S_h consists of S_s and S_d, where S_s includes candidate query graphs matching the predicted semantic structure, and S_d contains top-ranked graphs that do not conform to this structure, as defined in Eq. 1.

$$S_d = G_i | i \in [1, \xi \times m] \cap \mathbb{Z}, G_i \in S_C \setminus S_s \tag{1}$$

In Eq. 1, ξ controls the number of differing semantic structure graphs; m is the count of $S_C \setminus S_s$; and $\mathbb{Z}$ represents integers. S_d contains top $\xi \times m$ query graphs, and when ξ is 0, S_d is empty. Experiments show a nonempty S_d is beneficial, and the best graph in the refined set is used for answering the question.

4 Experimental Studies

4.1 Settings

Knowledge Base. We use Freebase as our knowledge base and, following the method of [7], we filter Non-English triples, leading to a total of 900M triples.

Questions. We utilized two datasets for performance evaluation. The first is ComplexQuestion [1], a set of 2,100 diverse, challenging questions divided into 1,300 for training and 800 for testing. The second dataset is WebQuestionSP [13], comprising 4,737 questions with a split of 3,098 for training and 1,639 for testing.

Baseline Methods. We used a list of models [2–7,9,10,12] as baseline methods, for performance comparison. † denotes the re-implementation by [7].

4.2 Results and Analysis

Overall Performance. Table 1 lists the F1 scores of our approach and other baseline methods. As can be seen, the F1 scores of our approach reach 74.2% and 43.2% on WebQuestionSP and ComplexQuestion, respectively. This indicates that our approach outperforms most state-of-the-art methods.

Performance of Sub-modules[2]. Our model exhibits superior performance, achieving an accuracy of 91.9% in topic entity recognition and an impressive F1

[2] Since ComplexQuestion does not provide ground truth, the performance evaluation of sub-modules is tested on WebQuestionSP only. For the submodule training on ComplexQuestion, we use the data provided by [4].

Table 1. Performance evaluation (F1)

Methods	WebQuestionSP (%)	ComplexQuestion (%)
[Luo *et al.*, 2018] [9]	-	42.8
[Chen *et al.*, 2018] [5] †	68.5	35.3
[Bhutani *et al.*, 2019] [2]	60.3	-
[Chen *et al.*, 2020] [4]	-	43.1
[Lan *et al.*, 2020] [7]	74.0	43.3
[Gu *et al.*, 2021] [6]	67.0	-
[Chen *et al.*, 2021] [3]	71.0	-
[Xie *et al.*, 2022] [12]	69.2	-
[Wang *et al.*, 2022] [10]	-	42.6
Ours	74.2	43.2

score of 97.5% in entity span detection. The efficacy of QGG is validated by a 91.34% DESM F1 score in generating the core path. The MSRM, maintaining constant parameters, stands out with a mean squared error of 46.56‰, leading to a final answering F1 score of 74.2%. The semantic structure classification model underscores an admirable F1 score of 86.23%, demonstrating successful optimization.

Influences by Parameters. We show the influences of parameters.

Varying k. As the beam size k increases, the overall quality of our query graphs improves. However, after $k = 3$, the rate of improvement slows, leading us to select $k = 3$ for an optimal balance of cost and performance.

Varying ξ and η. We find the following. (1) With the increase of ξ, *Avg-G* increases as well, which is as expected; while *Avg-G* is independent of η. (2) MSRM achieves the best performance when ξ is around 0.2 and 0.3 for WebQuestionSP and ComplexQuestion, respectively. (3) When $\xi = 0$, MSRM performs worst no matter which η is chosen. This shows that it is insufficient to only consider S_s when picking the best query graph. While when $\xi = 1$, MSRM performs worse than other ξ, indicating the effectiveness of our optimization technique. (4) When η equals to 0.5, MSRM achieves the best performance on both datasets.

5 Conclusion

In this paper, we propose a comprehensive approach, that is based the dual-encoder architecture to answering complex questions on knowledge bases. Extensive experiments on typical benchmark datasets show that: (1) our approach outperforms most existing methods; (2) the sub-modules perform quite well, *i.e.,* have higher F1 scores. This verifies that the dual-encoder architecture is able to improve the performance of complex KBQA.

Acknowledgments. This work is supported by Sichuan Scientific Innovation Fund, China (No. 2022JDRC0009).

References

1. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers. pp. 2503–2514 (2016)
2. Bhutani, N., Zheng, X., Jagadish, H.: Learning to answer complex questions over knowledge bases with query composition. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 739–748 (2019)
3. Chen, S., Liu, Q., Yu, Z., Lin, C.Y., Lou, J.G., Jiang, F.: Retrack: a flexible and efficient framework for knowledge base question answering. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, pp. 325–336 (2021)
4. Chen, Y., Li, H., Hua, Y., Qi, G.: Formal query building with query structure prediction for complex question answering over knowledge base. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (2020)
5. Chen, Z., Chang, C., Chen, Y., Nayak, J., Ku, L.: Uhop: an unrestricted-hop relation extraction framework for knowledge-based question answering. In: NAACL-HLT 2019, Minneapolis, MN, USA, vol. 1, pp. 345–356 (2019)
6. Gu, Y., Kase, S., Vanni, M., Sadler, B., Liang, P., Yan, X., Su, Y.: Beyond iid: three levels of generalization for question answering on knowledge bases. In: Proceedings of the Web Conference 2021, pp. 3477–3488 (2021)
7. Lan, Y., Jiang, J.: Query graph generation for answering multi-hop complex questions from knowledge bases. In: ACL 2020, Online, July 5–10, 2020 (2020)
8. Li, M., Ji, S.: Semantic structure based query graph prediction for question answering over knowledge graph. arXiv preprint arXiv:2204.10194 (2022)
9. Luo, K., Lin, F., Luo, X., Zhu, K.: Knowledge base question answering via encoding of complex query graphs. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2185–2194 (2018)
10. Wang, X., Luo, M., Si, C., Zhan, H.: Answering complex questions on knowledge graphs. In: KSEM 2022, Singapore, August 6–8, 2022, Proceedings, Part I. pp. 187–200. Springer (2022)
11. Wang, X., Yang, L., He, H., Fang, Y., Zhan, H., Zhang, J.: Enhanced simple question answering with contrastive learning. In: KSEM 2022, Singapore, August 6–8, 2022, Proceedings, Part I, pp. 502–515. Springer (2022). https://doi.org/10.1007/978-3-031-10983-6_39
12. Xie, M., Hao, C., Zhang, P.: A sequential flow control framework for multi-hop knowledge base question answering. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 8450–8460 (2022)
13. tau Yih, W., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. Meeting of the association for computational linguistics (2016)

Unsupervised Contrastive Learning of Sentence Embeddings Through Optimized Sample Construction and Knowledge Distillation

Yan Ding[1], Rize Jin[1](✉), Joon-Young Paik[1], and Tae-Sun Chung[2]

[1] School of Software, Tiangong University, Tianjin, China
jinrize@tiangong.edu.cn
[2] Department of Artifcial Intelligence, Ajou University, Suwon, South Korea

Abstract. Unsupervised contrastive learning of sentence embedding has been a recent focus of researchers. However, issues such as unreasonable division of positive and negative samples and poor data enhancement leading to text semantic changes still exist. We propose an optimized data augmentation method that combines contrastive learning's data augmentation with unsupervised sentence pair modelling's distillation. Our data augmentation uses in-sentence tokens for positive examples and text similarity for negative examples, while the distillation is conducted without supervised pairs. Experimental results on the STS task show that our method achieves a Spearman correlation of 81.03%, outperforming existing STS benchmarks.

Keywords: Contrastive Learning · Unsupervised Sentence Embedding · Distillation · Semantic Similarity

1 Introduction

Sentence representation is a vector with semantic information that represents sentences in natural language. The pre-trained language model BERT has been successful in many downstream NLP tasks. However, researchers have found that its performance on the STS task is not effective when directly using BERT embeddings [1]. This is due to the word vector representations in all layers of BERT are not isotropic and are unevenly distributed in direction [2]. To solve this problem, researchers have used Contrastive Learning by well-designed natural language augmentation methods.

However, there are still issues with the current models. Specifically, SimCSE [6] only uses dropout, so the positive samples are very similar, and there is a problem of learning saturation caused by feature suppression. When constructing negative examples, it is limited to using sentences in a batch as negative examples and ignores other sentences with similar semantics in the corpus. Trans-Encoder [3] exploits the advantages of both bi-encoder and cross encoder and

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 375–381, 2024.
https://doi.org/10.1007/978-981-99-7022-3_35

guides knowledge from them in an unsupervised manner to solve the unsupervised sentence pair modeling problem, but the important problem of obtaining the most suitable sentence representation is not fully resolved.

To address these issues, we propose a distillation-based model for unsupervised sentence vector representation learning. Our model allows our encoder to be trained on the positive and negative samples constructed by our data augmentation method. The main contributions of our study are as follows:

- We have integrated data augmentation for contrastive learning with the distillation approach used in unsupervised sentence pair modelling in a novel way. This combination has enabled us to leverage the strengths of both techniques and yield superior results in our model training.
- Our model requires no labels for training. For data augmentation, positive examples are generated by reusing in-sentence tokens, and negative examples are constructed by considering the text similarity of in-batch sentences. The distillation is conducted without supervised sentence pairs in an unsupervised manner.
- Our experimental results demonstrate that our sophisticated data augmentation improves the performance with the distillation orthogonally. Our model achieves state-of-the-art performance on the STS benchmark test among 'BERT-based' models in an unsupervised setting.

2 Distillation-Based Unsupervised Sentence Representation Learning

2.1 Model Architecture

Figure 1 illustrates our model architecture, which mainly consists of three parts: base-encoder, bi-encoder, and knowledge distillation.

The base-encoder uses a contrastive learning framework. The sentence after the subword repetition of the input sentence is used as a positive example, and other randomly sampled sentences in the same batch are used as negative examples. GS-InfoNCE is used as a contrastive learning loss function.

The bi-encoder calculates the similarity score between the input sentence and other sentences in the corpus, and ranks them according to the score to obtain a ranking vector. By calculating the inner product of the ranking vectors, we can obtain their similarity score s_{ij}. We calculate the cross-entropy loss between s_{ij} and the similarity score obtained by the base-encoder.

In knowledge distillation, the pre-trained language model BERT and the bi-encoder are each other's teacher and each other's students, and use the similarity score of each other's marks as labels for knowledge distillation.

2.2 Data Augmentation Strategy

Subword Id Self-replication. We use subword repetition as a data augmentation strategy to construct positive examples corresponding to the input text.

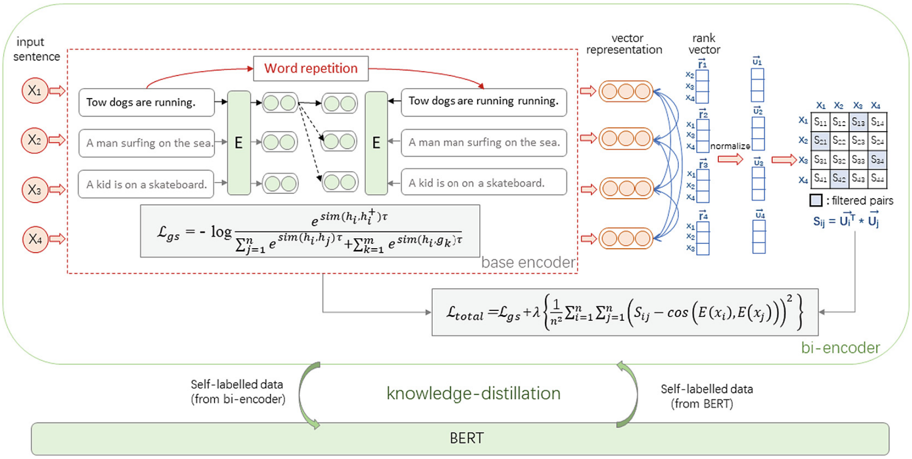

input sentence
X1
X2
X3
X4
Word repetition
Tow dogs are running.
A man surfing on the sea.
A kid is on a skateboard.
E
E
Tow dogs are running running.
A man man surfing on the sea.
A kid is on on a skateboard.
$\mathcal{L}_{gs} = -\log \frac{e^{sim(h_i,h_i^+)\tau}}{\sum_{j=1}^{n} e^{sim(h_i,h_j)\tau} + \sum_{k=1}^{m} e^{sim(h_i,g_k)\tau}}$
base encoder
vector representation
rank vector
$\vec{r_1}$
$\vec{r_2}$
$\vec{r_3}$
$\vec{r_4}$
normalize
$\vec{u_1}$
$\vec{u_2}$
$\vec{u_3}$
$\vec{u_4}$
S11 S12 S13 S14
S21 S22 S23 S24
S31 S32 S33 S34
S41 S42 S43 S44
□ : filtered pairs
$S_{ij} = \vec{U_i}^T * \vec{U_j}$
$\mathcal{L}_{total} = \mathcal{L}_{gs} + \lambda \left\{ \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(S_{ij} - cos\left(E(x_i), E(x_j)\right)\right)^2 \right\}$
bi-encoder
Self-labelled data (from bi-encoder)
knowledge-distillation
Self-labelled data (from BERT)
BERT

Fig. 1. Architecture of distillation-based unsupervised sentence representation learning.

Convert the input text into an id list $S = \{s_1, s_2, s_3, \ldots, s_n\}$ through the tokenization, use uniform distribution to randomly select 1 or 2 ids in the id sequence for repetition, and obtain the id sequence after subword repetition , such as $S' = \{s_1, s_2, s_2, s_3, \ldots, s_n\}$. Then S and S' are sent to the Encoder with the same dropout probability, and two different representations $\overrightarrow{h}$ and $\overrightarrow{h^+}$ corresponding to similar semantics are obtained, which are used as positive sample pairs after the contrastive learning data expansion.

Text Similarity Score Ranking. We reconstruct the negative examples of the input text by ranking the samples by their similarity scores. First we send all sentences in the same batch to the base-encoder to get the corresponding vector representation. We use the vector representation h_i of the input sentence x_i and the vector representation $\{h_1, h_2, h_3, \ldots, h_n\}$ of other sentences in the same batch to calculate the cosine similarity score, and make a similarity ranking for other sentences according to the score (similarity score from high to low), such as $\{x_3, x_1, x_n, \ldots, x_2\}$. Next, the rank vector r_i of the input sentence x_i is obtained, such as $[3, 1, n, \ldots, 2]$, and each value in the rank vector r_i is normalized to $-1 \sim 1$, and the result is represented by u_i . Finally, do the inner product for all u_i to calculate another representation of the similarity score s_{ij}.

We filter out s_{ij} that are too high and too low $(0.5 < s_{ij} < 1)$. Finally, we calculate the cross-entropy loss, reducing the KL divergence between base-encoder and bi-encoder. At this point, the loss function of the bi-encoder is:

$$\begin{aligned} L_{total} &= L_{gs} + \lambda\{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(S_{ij} - cos(E(x_i), E(x_j)))^2\} \\ &= -log\frac{e^{sim(h_i, h_i^+)\tau}}{\sum_{j=1}^{n} e^{sim(h_i, h_j)\tau} + \sum_{k=1}^{m} e^{sim(h_i, g_k)\tau}} \\ &+ \lambda\{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(S_{ij} - cos(E(x_i), E(x_j)))^2\} \end{aligned} \tag{1}$$

2.3 Knowledge Distillation

In the distillation process, BERT and the bi-encoder are teachers and students of each other, teaching and learning. Specifically, when the bi-encoder is used as the teacher model, the cosine similarity score obtained by the sentence pair through the bi-encoder is used as the training label of the student model BERT. Conversely, when BERT is used as the teacher model, the scalars mapped by BERT's [CLS] representation are used as the training labels of the student model bi-encoder.

In every distillation process, the predicted value and the real value are used to calculate the mean square error loss and minimize the two KL divergence between scores. The loss function of the model in this process is:

$$L_{MSE} = -\frac{1}{N}\sum_{n=1}^{N}(\theta_n - \phi_n)^2 \tag{2}$$

According to the method described above, the knowledge distillation process is iteratively repeated, and finally a bi-encoder with stronger sentence pair scoring ability isobtained.

3 Experiments

3.1 Datasets

We randomly sampled 1 million pieces of data on English Wikipedia to train the model in an unsupervised manner, and used the STS dataset[1] to evaluate the ability of the model to measure the semantic similarity of sentences, including 7 subtasks, namely STS12-16, STS-B (STS Benchmark) and SICK-R (SICK-Relatedness). This dataset is the most widely used benchmark dataset for evaluating unsupervised sentence embedding tasks. Each sentence pair is given a score of 0–5 by humans based on semantic similarity.

[1] https://github.com/dingyan0352/dyfinalcode.

3.2 Results and Discussion

We compare our method with current heuristic and advanced unsupervised sentence vector representation models, such as post-processing methods BERT_flow [4], BERT_whitening [8], contrastive learning methods ConSERT [5], SimCSE [6], DCLR [9], SNCSE [10], SimCSE+RankEncoder [7], Trans-Encoder [3]. The experimental results are shown in Table 1.

Table 1. Evaluation results of sentence vector representations on the STS task.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT_flow(2020)†	63.48	72.14	68.42	73.77	75.37	70.72	63.11	69.57
BERT_whitening(2021)†	63.89	73.76	69.08	74.59	74.40	71.43	62.20	69.90
ConSERT(2021)†	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE(2021)†	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DCLR(2022)†	70.81	83.73	75.11	82.56	78.44	78.31	71.59	77.22
SimCSE+RankEncoder(2022)†	75.00	82.00	75.20	83.00	79.80	80.40	71.10	78.10
Trans-Encoder(2022)†	72.17	84.40	76.69	83.28	80.91	81.26	71.84	78.65
SNCSE(2022)†	70.67	84.79	76.99	83.69	80.51	81.35	**74.77**	78.97
ours	**79.14**	**85.13**	**78.59**	**85.14**	**81.90**	**83.31**	74.03	**81.03**

Table 1 shows the STS performance of our method on 7 STS datasets and their average performance in the unsupervised setting. The experiments are based on BERT_base, and the results with † are all from the original paper. We report the Spearman correlation coefficient between the similarity scores annotated by human annotators and those predicted by the model. With the exception of SICK-R, we achieved the best results on every single dataset.

On the SICK-R dataset, our method also improves a lot compared to previous models, but does not surpass the SNCSE score. Because the SICK-R dataset labels the relationship between two sentences: implication, contradiction and neutrality. It may be due to the fact that SICK-R contains more contradictory pairs, resulting in the model not being able to learn more similarities with the input sentences in the process of ranking the sentences in the corpus, so the results are not achieving the best compared with other datasets.

3.3 Ablation Study

To verify the effectiveness of our proposed method and explore better hyperparameter settings, we conduct ablation studies on the STS-B validation set to evaluate the model.

Effect of Hyperparameters on Bi-encoder Loss. In the model training, we introduced the main hyperparameter λ of the loss function L_{total} to balance the weight of the base-encoder and the bi-encoder. In the experiment, we found that $\lambda = 0.05$ is the best value (Table 2).

Table 2. Spearman score of STS-B validation set under different λ settings.

λ	0.03	0.04	0.05	0.06	0.07
STS-B	85.98	85.42	**86.91**	85.57	85.28

Effect of Data Enhancement and Knowledge Distillation. We augment the baseline, SimCSE, with the proposed data enhancement methods and knowledge distillation strategy: GS-InfoNCE, subword id self-replication, text similarity score ranking and knowledge distillation strategy. We investigate the effect of combinations of these methods on performance. Table 3 shows that each method contributes the performance improvement, demonstrating the effectiveness of our methods.

Table 3. The Spearman score of the STS-B validation set in the SimCSE-based comparison test, the results with † are from the original paper.

Model	STS-B
SimCSE †	82.5
+Gaussian noise	83.45
+Gaussian noise+Text similarity ranking	84.34
+Gaussian noise+Text similarity ranking+Repeat subword id	84.72
+Gaussian noise+Text similarity ranking+Repeat subword id+Knowledge distillation	**86.91**

4 Conclusion

Our proposed unsupervised sentence embedding method addresses the limitations of BERT sentence vectors and significantly improves the representation ability of sentence embeddings. We construct data augmentation strategies such as subword id self-replication and text similarity score ranking and we also employ a pre-trained language model BERT and bi-encoder for knowledge distillation in an unsupervised manner. The experimental results demonstrate that our method achieves outstanding performance on the STS benchmark test, surpassing other 'BERT-based' models in an unsupervised environment. In future work, we plan to explore sample construction methods based on generative language models and assess the method's transfer and generalization performance for downstream tasks.

References

1. Pennington, J., Socher, R., Manning, C. D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543. ACL Press, Doha (2014)

2. Ethayarajh, K.: How contextual are contextualized word representations. comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 55–65. ACL Press, Hong Kong (2019)
3. Liu, F., Jiao, Y., Massiah, J., Yilmaz, E., Havrylov, S.: Trans-Encoder: unsupervised sentence-pair modelling through self-and mutual-distillations. In: Proceedings of ICLR 2022, Louisiana (2022)
4. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 9119–9130. ACL Press, Online (2020)
5. Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., Xu, W.: Consert: a contrastive framework for self-supervised sentence representation transfer. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 5065–5075. ACL Press, Online (2021)
6. 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, Online and Punta Cana, pp. 6894–6910. ACL Press, Online and Punta Cana (2021)
7. Seonwoo, Y., et al.: Ranking-Enhanced Unsupervised Sentence Representation Learning. arXiv preprint arXiv:2209.04333 (2022)
8. Su, J., Cao, J., Liu, W., Ou, Y.: Whitening sentence representations for better semantics and faster retrieval. arXiv preprint arXiv:2103.15316 (2021)
9. Zhou, K., Zhang, B., Zhao, W.X., Wen, J.: Debiased contrastive learning of unsupervised sentence representations. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 6120–6130. ACL Press, Dublin (2022)
10. Wang, H., Li, Y., Huang, Z., Dou, Y., Kong, L., Shao, J.D.: SNCSE: Contrastive Learning for Unsupervised Sentence Embedding with Soft Negative Samples. arXiv preprint arXiv:2201.05979 (2022)

Optimization

Automatically Choosing Selection Operator Based on Semantic Information in Evolutionary Feature Construction

Hengzhe Zhang[1], Qi Chen[1](✉), Bing Xue[1], Wolfgang Banzhaf[2], and Mengjie Zhang[1]

[1] Centre for Data Science and Artificial Intelligence and School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{hengzhe.zhang,qi.chen,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz

[2] Department of Computer Science and Engineering, Michigan State University, East Lansing 48824, USA
banzhafw@msu.edu

Abstract. In recent years, genetic programming-based evolutionary feature construction has shown great potential in various applications. However, a critical challenge in applying this technique is the need to select an appropriate selection operator with great care. To tackle this issue, this paper introduces a novel approach that leverages the Thompson sampling technique to automatically choose the optimal selection operator based on semantic information of genetic programming models gathered during the evolutionary process. The experimental results on a standard symbolic regression benchmark containing 37 datasets show that the proposed adaptive operator selection algorithm outperforms expert-designed operators, demonstrating the effectiveness of the adaptive operator selection algorithm.

Keywords: Genetic Programming · Evolutionary Feature Construction · Adaptive Operator Selection

1 Introduction

Automated feature construction is an important technique in the machine learning domain and has achieved significant success in various applications [1,2]. Formally, given a dataset $\{X, Y\}$, the objective of automated feature construction is to develop a set of features $\Phi_1(X), \ldots, \Phi_m(X)$ that enhance the performance of a learning algorithm on the given dataset. The effectiveness of automated feature construction techniques has been well-demonstrated by deep learning [2]

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-99-7022-3_36.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 385–397, 2024.
https://doi.org/10.1007/978-981-99-7022-3_36

and kernel methods [3]. However, the interpretability of the constructed features remains a notable point of criticism within the field, demanding further investigation and deliberation [4].

In recent years, interpretable automated feature construction techniques, particularly those based on genetic programming (GP), have demonstrated impressive performance for enhancing ensemble learning algorithms [5,6], compared to learning with original features. The variable-length representation and gradient-free search mechanism make GP suitable for exploring flexible high-order features, like $(x_1 + x_2) * x_3$, on non-differentiable machine learning algorithms. Based on the evaluation methods, evolutionary feature construction methods can be categorized into filter-based [7], wrapper-based [1,8], and embedded methods [9,10]. Filter-based methods do not rely on any specific machine learning algorithm, making them efficient and generalize well to different learning algorithms [7]. On the other hand, wrapper-based methods evaluate the constructed features on a specific learning algorithm, which may lead to better features at the cost of higher computation time [8]. Finally, embedded methods integrate the feature construction into model learning, with GP-based symbolic regression being a representative example [9].

To improve the search effectiveness, numerous selection operators have developed for GP, which are used in GP to select promising individuals for crossover and mutation to generate new solutions, playing a crucial role in driving the evolutionary progress. Representative examples include standard tournament selection [11], clustering tournament selection [12], lexicase selection [13], and multi-dimensional archive of phenotypic elites (MAP-Elites) [14]. Each of these operators demonstrates unique advantages in different scenarios, such as dynamic selection pressure adjustment [12], specialist preservation [15], and diversity enhancement for ensemble learning [14]. However, selecting the most appropriate selection operators in real-world tasks is challenging because suitable operators vary with different optimization landscapes or phases, often unknown in advance. Recent work has shown that GP performs well using tournament selection for the first 10% of generations, then lexicase selection for the rest [16]. Therefore, an adaptive operator selection (AOS) algorithm for the selection operator is needed.

There are two potential approaches for automatic operator selection. First, operators can be selected based on historical knowledge [17], also known as algorithm recommendation. However, obtaining historical knowledge requires running numerous experiments in advance. Furthermore, the best operator may change during the evolutionary process. Therefore, adaptive operator selection may be a better choice [18]. Given the success of AOS techniques in selecting genetic operators for continuous numerical optimization problems [19–21], particularly AOS based on the multi-armed bandit and dynamic Thompson sampling [21], this paper explores the feasibility of automatically selecting the optimal selection operators during evolution. However, in GP, relying only on the improvement of fitness values may not provide sufficient rewards to selection operators. Thus, this paper explores the use of GP semantics to design an effective AOS method, where the semantics of each GP program refers to the output values of each GP individual [22].

Goals: The main goal of this paper is to develop an AOS method for determining selection operators in GP-based feature construction. The specific objectives of this work are as follows:

1. Developing a portfolio of selection operators for AOS in evolutionary feature construction.
2. Proposing a semantic-based AOS method using dynamic Thompson sampling to adaptively determine an appropriate selection operator during the evolutionary process.
3. Evaluating the effectiveness of different selection operators and credit assignment strategies on 37 datasets.

2 Related Work

2.1 Multi-armed Bandit

The multi-armed bandit is a reinforcement learning technique that aims to balance the exploration and exploitation of different options, also known as "arms", based on past rewards [21,23,24]. In the context of GP, selection operators can be considered arms. In each generation, an operator with the highest estimated rewards is chosen, and applying this operator to select two parents is a trial. The goal is to find the optimal selection operator for GP through trials. Numerous multi-armed bandit algorithms have been developed for various scenarios, and two key techniques are particularly useful for GP.

- Dynamic Multi-armed Bandit [23,24]: In GP, the optimal selection operator may change during the evolution process. Therefore, the multi-armed bandit algorithm should have the ability to forget long-term history and focusing on recent knowledge in order to adapt to these changes, which is known as the dynamic multi-armed bandit. This is achieved via a forgetting mechanism through explicit drift detection algorithm [24] or simple decay over time [21].
- Thompson Sampling [21]: GP is a population-based optimization algorithm, and it requires to have a sampling algorithm that can generate multiple trials of selection operators simultaneously. Thus, it is desirable to have an explicit reward distribution for each selection operator, and each trial can sample a value from each distribution to determine which selection operator to choose. This process is known as Thompson sampling. Compared to using the upper confidence bound and expected improvement, Thompson sampling allows for trying different selection operators in each round, which is more naturally suited for GP.

2.2 Automatic Operator Selection

In the evolutionary computation domain, numerous genetic operators have been developed, and studies have shown that combining the advantages of different

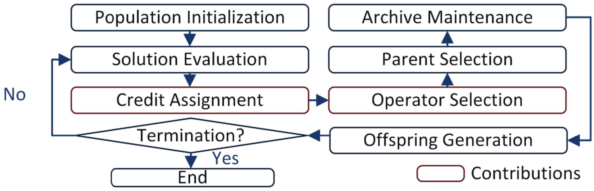


Fig. 1. Workflow of the proposed algorithm.

operators is beneficial for addressing optimization problems [25]. Instead of simple hybridization, automatic operator selection has become a hot topic in the evolutionary computation (EC) domain, aiming to dynamically choose the optimal operator at each stage [20]. One pioneering approach is probability matching, which adjusts the probability of selecting each individual based on reward distribution [19], where the reward is typically defined as the improvement in fitness values, either in a real-valued form [18] or a boolean-valued form [21]. However, probability matching does not consider accumulative reward distribution. To address this limitation, adaptive purist was proposed to accumulate reward during the evolutionary process, leading to improved operator selection performance [19]. Building upon this idea, a dynamic multi-armed bandit algorithm with the Page-Hinkley test was proposed to further enhance operator selection effectiveness [23,24]. Under the framework of fitness-rate-rank-based multi-armed bandit (FRRMAB) [20], dynamic Thompson Sampling [21], deep reinforcement learning [18,26], and other methods have been developed. While numerous approaches have been proposed for automatic operator selection, most of them primarily focus on solving numerical optimization problems and emphasize the selection of genetic operators. For GP-based feature construction algorithms, the effectiveness of automatic operator selection methods for selection operators still requires further investigation.

3 The New Algorithm

3.1 Model Representation

In this paper, we focus on evolutionary feature construction for a linear regression model due to its simplicity and effectiveness. Specifically, each GP individual consists of m GP trees, representing m constructed features $\phi_1, \ldots, \phi_m$. Based on these constructed features, a linear model is trained to make predictions for the given data. To ensure accurate and robust predictions, the final predictions are made by an ensemble model that incorporates the top-$|A|$ individuals obtained during the evolutionary process, where $|A|$ is an algorithm parameter referring to the ensemble size.

3.2 Algorithm Framework

The algorithm follows a general framework of GP, as illustrated in Fig. 1, where credit assignment and operator selection are the key components for the new AOS method to determine the best selection operator. The main components of the proposed algorithm are described as follows:

- Population Initialization: During the initialization stage, GP trees are initialized using the ramped half-and-half method [11]. Specifically, each GP individual with m trees is randomly generated, with each tree representing a constructed feature.
- Solution Evaluation: In the evaluation stage, all GP individuals are evaluated using ridge regression. Specifically, m trees construct m features, and these constructed features are then fed into a linear model to make predictions for the given data. Fitness is determined by the R^2 score on training data, with leave-one-out cross-validation to avoid overfitting by selecting a regularization coefficient from $\{0.1, 1, 10\}$.
- Credit Assignment: This phase updates the reward distribution of operators based on evaluation results. The details of the credit assignment are presented in Sect. 3.4.
- Operator Selection: Selection operators are sampled to select pairs of individuals for crossover and mutation. For a population of n individuals, it needs to sample $\frac{n}{2}$ operators. In this paper, lexicase selection and tournament selection are defined as candidate operators since they are commonly used in GP.
- Parent Selection: At this stage, GP individuals are selected using the $\frac{n}{2}$ sampled selection operators to select promising individuals.
- Archive Maintenance: In addition to selecting offspring, the top individuals in the population and the archive A are compared, and the top $|A|$ individuals are stored in the archive to form an ensemble model.
- Offspring Generation: Offspring generation is a stage where new GP individuals are generated using random subtree crossover and guided subtree mutation operator [6]. In this paper, each individual has m GP trees, and thus genetic operators are invoked m times for each individual to ensure sufficient variations.

3.3 Selection Operators

This paper considers two widely used selection operators:

1. Tournament Selection: The tournament selection operator randomly samples t individuals from the population, where t is the tournament size, and selects the best as the parent. Here, $t = 7$ is used according to common settings in GP literature.
2. Lexicase Selection [13]: The lexicase selection operator iteratively constructs filters to progressively narrow down the selection pool until one individual remains. In each round, the filter is constructed as $min_{\Phi \in P} \mathcal{L}_k(\Phi) + \epsilon_k$, with ϵ_k as the median absolute deviation of the loss on the k-th instance among all individuals $\Phi \in P$.

Intuitively, tournament selection tends to converge by favoring individuals with higher overall fitness values. In contrast, lexicase selection emphasizes improving fitness on different instances, thus promoting diversity. Thus, using an AOS method to choose between these can simultaneously improve the overall accuracy and the accuracy on tough instances, leading to a superior ensemble model. This idea is inspired by AdaBoost, where some learners have good overall accuracy while others focus on hard instances. The ensemble model can then achieve good accuracy on all instances.

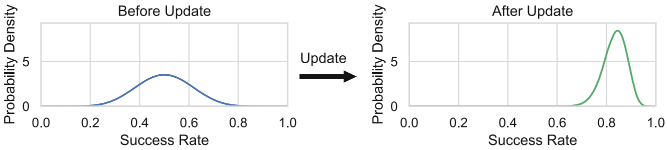


Fig. 2. Credit assignment update in multi-armed bandit using beta distribution.

3.4 Dynamic Multi-armed Bandit

To apply the dynamic multi-armed bandit in GP, two components, credit assignment and operator selection, must be carefully designed: credit assignment allocates rewards to each operator, and operator selection samples operators based on estimated rewards. This section delves into these components.

Credit Assignment: Credit assignment is a stage where rewards are assigned to each selection operator, involving two main questions:

- **How to define a successful trial?** In this paper, a successful trial for a selection operator is defined as having any improvement in one dimension of the semantics compared to the best semantics among all parents. Semantics $(\Phi(X_1), \ldots, \Phi(X_N))$ refer to the output values of each GP individual Φ, which can determine a loss vector $(\mathcal{L}_{\Phi,1}, \ldots, \mathcal{L}_{\Phi,N})$. At each generation, all individuals in the population $\Phi \in P$ can collectively form the best loss vector, where each element corresponds to the minimum loss value that individuals in P achieve on each training instance. This vector is denoted as $\{\min_{\Phi \in P} \mathcal{L}_{\Phi,i} | i \in [1, N]\}$. For a new individual Φ^+, if $\exists_{i \in [1,N]} \mathcal{L}_{\Phi^+,i} < \min_{\Phi \in P} \mathcal{L}_{\Phi,i}$, it is considered a successful trial and is rewarded with one point. This reward strategy is based on the principle that if a new individual outperforms all existing individuals on a data sample, it indicates that useful knowledge has been discovered, allowing the new individual to achieve the best performance on that sample, even if average fitness does not increase.
- **How to update estimated reward distribution?** As shown in Fig. 2, in order to use Thompson sampling, $k = 2$ beta distributions $\theta = (\theta_1, \ldots, \theta_k)$ are defined for k selection operators with two sets of parameters $\alpha = (\alpha_1, \ldots, \alpha_k)$ and $\beta = (\beta_1, \ldots, \beta_k)$. All these parameters are initialized to one. After obtaining a successful trial for each operator, the α parameters are updated, i.e.,

$\alpha_i = \alpha_i + 1$ and $\beta_i = \beta_i$. Otherwise, if the trial is unsuccessful, the β parameters are updated, i.e., $\alpha_i = \alpha_i$ and $\beta_i = \beta_i + 1$. Due to the changing dynamics of the evolutionary process, the reward distribution for k operators may change. Therefore, weight decay is applied to all distributions. After each round of updating, the distribution parameters α, β are decayed by a decay factor γ, which is set to 0.9 in this paper. In order to prevent the probability from diminishing to an extremely low value, which could lead to an operator never being chosen in the future, the decayed value is restricted to a minimum of 1.

Operator Selection: Once the parameters of all selection operators have been updated, the selection operators are sampled based on the probabilities associated with each selection operator in the operator selection stage. Specifically, the probability of choosing selection operator i is defined in Eq. (1) [21], where $\Gamma(x)$ is the gamma function, that is, $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

$$\mathcal{P}^{\text{Beta}}\left(\theta_i\right) = \frac{\Gamma\left(\alpha_i + \beta_i\right)}{\Gamma\left(\alpha_i\right)\Gamma\left(\beta_i\right)} \theta_i^{\alpha_i - 1}\left(1 - \theta_i\right)^{\beta_i - 1}, \tag{1}$$

After applying the selected operator to select an individual, the selected operator is marked associated with the selected individual to be able to make credit assignments.

Table 1. Parameter settings for GP.

Parameter	Value
Maximal Population Size	30D (500)
Number of Generations	200
Ensemble Size	30
Crossover and Mutation Rates	0.9 and 0.1
Maximum Tree Depth	10
Initial Tree Depth	0–2
Number of Trees in An Individual	10
Elitism (Number of Individuals)	1
Functions	Add, Sub, Mul, AQ, Sin, Cos, Abs, Max, Min, Negative

4 Experimental Settings

4.1 Experimental Dataset

The experimental datasets are obtained from the Penn Machine Learning Benchmark (PMLB) [27] [1]. Due to the constraints of computational resources, we

[1] Details of Datasets: https://epistasislab.github.io/pmlb/

selected datasets with fewer than 5000 instances. Additionally, we only evaluate performance on real-world datasets. Based on these criteria, 37 datasets are finally selected. Specifically, the number of instances in these datasets ranges from 47 to 3848, and the number of dimensions falls between 2 and 124.

4.2 Parameter Settings

The parameters follow the conventions established in the GP literature, as outlined in Table 1. The population size is set to 30 times the number of original features, with a maximum limit of 500. To address the issue of zero-division errors, we replace the division operator with the analytical quotient operator [28]. The analytical quotient operator is defined as $AQ\left(a,b\right)=\frac{a}{\sqrt{1+b^2}}$, where a and b are two parameters.

4.3 Evaluation Protocol

The experiments are conducted on the New Zealand e-science infrastructure (NeSI), which consists of a cluster of AMD EPYC 7713 CPUs. For the evaluation protocol, each algorithm is independently tested on each dataset for 30 runs. The comparisons between algorithms are performed using the Wilcoxon signed-rank test. For each run, the datasets are split into training and test sets in an 80:20 ratio. The performance of an algorithm is evaluated using the R^2 score as the performance metric based on the test set.

4.4 Baseline Algorithms

This work considers three baseline selection operators within GP-based feature construction algorithms:

- Lexicase [13]: Only the automatic epsilon lexicase selection operator is used in GP.
- Tournament: Only the tournament selection operator is used in GP.
- TR/LS [16]: TR/LS is a heuristic operator selection strategy designed by GP experts. Tournament selection is used in the first q generations to avoid hyper-selection, and lexicase selection is used in the remaining generations. In the original paper of TS/LS [16], q is set to 10% of the total generations. Therefore, q is set to 10 in this paper.

Moreover, two different credit assignment strategies are studied to determine the best one for GP:

- Semantics: Any improvement achieved by the selection operator over a value in the vector of squared errors of parents is considered a successful improvement. This is the credit assignment strategy used in this paper.
- Fitness: Any improvement achieved by the selection operator over the best R^2 score of parents is considered a successful improvement.

5 Experimental Results

5.1 Comparison Between Selection Operators

Test Score: The experimental results using different selection operators are presented in Table 2 [2]. The results demonstrate that AOS significantly outperforms tournament selection and lexicase selection operators on 16 and 9 datasets, respectively, while not performing worse on any dataset. These results highlight the advantages of combining different selection operators in the evolutionary process. Interestingly, AOS also outperforms TL/LS, a hybrid operator designed by GP experts. As shown in Table 2, AOS performs better than the TR/LS operator on 6 datasets, similar on 30 datasets, and worse on only one dataset. These results suggest that AOS can provide an advantage over the heuristic operator selection strategy designed by GP experts.

Table 2. Comparison of R^2 scores for different selection operators.

	TR/LS	Tournament	Lexicase
AOS	6(+)/30(∼)/1(-)	16(+)/21(∼)/0(-)	9(+)/28(∼)/0(-)
TR/LS	—	11(+)/25(∼)/1(-)	5(+)/30(∼)/2(-)
Tournament	—	—	0(+)/26(∼)/11(-)

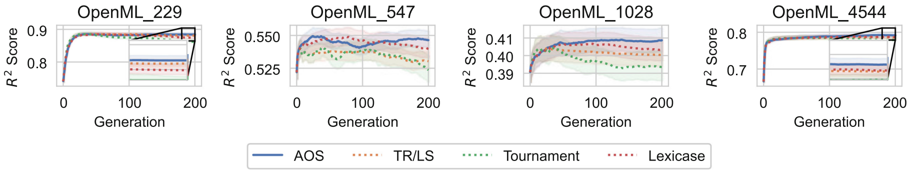


Fig. 3. Evolutionary plots of test R^2 scores using four different selection operators.

To gain further insights into the advantage of using AOS over deterministic operators, we plot the curve of test R^2 scores for representative datasets in Fig. 3. The results demonstrate that AOS can improves test R^2 scores in later generations, whereas tournament selection operators suffer from severe overfitting, leading to degraded test R^2 scores in later generations. Therefore, in the following sections, we focus on analyzing the reasons behind the improved generalization ability of the ensemble models made by AOS.

[2] Detailed Results: https://tinyurl.com/AOS-GP-Supplementary-Material

Operator Selection Patterns: To understand why AOS outperforms TS/LS, lexicase selection and tournament selection, we analyze the selection ratios of different operators during the evolutionary process on four datasets, as shown in Fig. 4. The results indicate that the lexicase selection operator performs well and is selected more frequently than the tournament selection operator. For instance, on the "OpenML_547" dataset, the lexicase selection operator has selected an average of 181 times at the last generation, while the tournament selection operator is selected only 29 times on average. Although the proportion of tournament selection operators is relatively small compared to the lexicase selection operator, this small proportion is not negligible considering the significant improvements achieved with using AOS compared with using lexicase selection alone, as presented in Fig. 3.

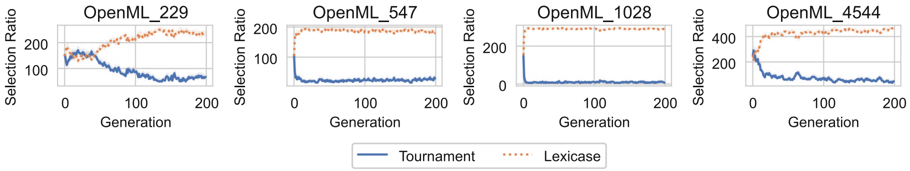


Fig. 4. Selection ratios of different operators during the evolutionary process.

Cosine Distance: To further demonstrate the reasons behind the superior performance of AOS, we introduce the average cosine distance. The average cosine distance is used as a metric to measure the complementarity of different models in the archive, which is crucial for ensemble learning [14]. A larger cosine distance indicates greater complementarity. The results in Fig. 5 demonstrate that the adaptive selection operator achieves the greatest cosine distance by adaptively balancing lexicase and tournament selections. Although lexicase selection fosters a high level of diversity, incorporating a small proportion of tournament selection appears to enhance it further. This may be because lexicase selection can suffer from hyper-selection [29], where a superior individual dominating the other individuals can be chosen up to 90% of the time [29]. In such cases, introducing a moderate proportion of tournament selection may improve archive diversity. In other cases, the high usage of lexicase selection ensures a high level of population diversity for discovering well-performing models on different training instances, thereby forming a strong ensemble learning model.

5.2 Comparison of Credit Assignment Strategies

To demonstrate the superiority of the proposed credit assignment strategy, this section compares the effectiveness of two different credit assignment strategies for GP. The comparison results between semantics-based credit assignment and fitness-based credit assignment are presented in Fig. 6a. The results indicate that utilizing semantic information for credit assignment leads to significantly better

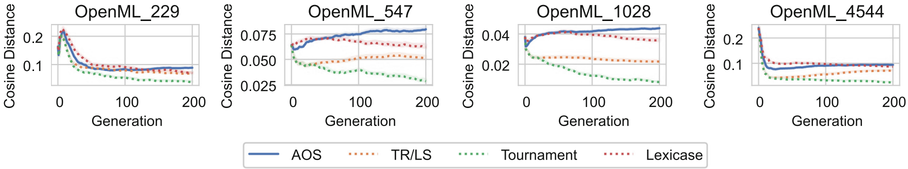
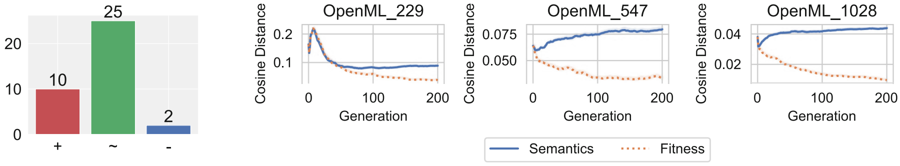


Fig. 5. Cosine distance of archived individuals.

results on 10 datasets while performing worse on 2 datasets. This performance can be attributed to the fact that credit assignment based on semantics encourages the discovery of solutions with diverse semantics, thereby facilitating the formation of a high-quality ensemble model. The evolutionary plots of the cosine distance of archived individuals are shown in Fig. 6b, clearly demonstrating the advantage of the semantic-based credit assignment strategy in terms of diversity maintenance.



(a) Test R^2 scores.

(b) Cosine distance.

Fig. 6. (a). Statistical comparison of test R^2 scores. ("+"/red bar indicates that for a dataset, the semantic-based credit assignment strategy outperforms the fitness-based credit assignment strategy.) (b). Evolutionary plots of cosine distances. (Color figure online)

6 Conclusions

This work aims to automate the determination of the optimal selection operator during the process of evolutionary feature construction. To achieve this, we use the Thompson sampling technique to sample selection operators based on their estimated rewards, where the reward is defined as an improvement in semantics. The experimental results on 37 datasets demonstrate that the proposed method outperforms using sole lexicase selection, sole tournament selection and a manually designed hybrid selection operator, highlighting the advantages of employing AOS. However, it should be noted that this paper is limited to the use of AOS for determining selection operators. In future research, it would be valuable to extend this framework to the selection of genetic operators and environmental selection operators in order to further enhance the performance.

References

1. La Cava, W., Singh, T.R., Taggart, J., Suri, S., Moore, J.H.: Learning concise representations for regression by evolving networks of trees. In: ICLR (2018)
2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
3. Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Netw. **12**(2), 181–201 (2001)
4. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5), 206–215 (2019)
5. Zhang, H., Zhou, A., Zhang, H.: An evolutionary forest for regression. IEEE Trans. Evol. Comput. **26**(4), 735–749 (2022)
6. Zhang, H., Zhou, A., Chen, Q., Xue, B., Zhang, M.: SR-Forest: a genetic programming based heterogeneous ensemble learning method. IEEE Trans. Evol, Comput (2023)
7. Neshatian, K., Zhang, M., Andreae, P.: A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. IEEE Trans. Evol. Comput. **16**(5), 645–661 (2012)
8. Virgolin, M., Alderliesten, T., Bosman, P.A.: On explaining machine learning models by evolving crucial and compact features. Swarm Evol. Comput. **53**, 100640 (2020)
9. Chen, Q., Zhang, M., Xue, B.: Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Trans. Evol. Comput. **21**(5), 792–806 (2017)
10. Zhang, H., Zhou, A., Qian, H., Zhang, H.: PS-Tree: a piecewise symbolic regression tree. Swarm Evol. Comput. **71**, 101061 (2022)
11. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Stat. Comput. **4**(2), 87–112 (1994)
12. Xie, H., Zhang, M.: Parent selection pressure auto-tuning for tournament selection in genetic programming. IEEE Trans. Evol. Comput. **17**(1), 1–19 (2012)
13. La Cava, W., Helmuth, T., Spector, L., Moore, J.H.: A probabilistic and multi-objective analysis of lexicase selection and ε-lexicase selection. Evol. Comput. **27**(3), 377–402 (2019)
14. Zhang, H., Chen, Q., Tonda, A., Xue, B., Banzhaf, W., Zhang, M.: Map-elites with cosine-similarity for evolutionary ensemble learning. In: EuroGP. pp. 84–100. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-29573-7_6
15. Helmuth, T., Pantridge, E., Spector, L.: Lexicase selection of specialists. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1030–1038 (2019)
16. Xu, M., Mei, Y., Zhang, F., Zhang, M.: Genetic programming with lexicase selection for large-scale dynamic flexible job shop scheduling. IEEE Trans. Evol., Comput. (2023)
17. Tian, Y., Peng, S., Zhang, X., Rodemann, T., Tan, K.C., Jin, Y.: A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. IEEE Trans. Artif. Intell. **1**(1), 5–18 (2020)
18. Tian, Y., Li, X., Ma, H., Zhang, X., Tan, K.C., Jin, Y.: Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. IEEE Trans. Emerg. Top. Comput, Intell (2022)

19. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: GECCO, pp. 1539–1546 (2005)
20. Li, K., Fialho, A., Kwong, S., Zhang, Q.: Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **18**(1), 114–130 (2013)
21. Sun, L., Li, K.: Adaptive operator selection based on dynamic Thompson sampling for MOEA/D. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12270, pp. 271–284. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58115-2_19
22. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 21–31. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32937-1_3
23. DaCosta, L., Fialho, A., Schoenauer, M., Sebag, M.: Adaptive operator selection with dynamic multi-armed bandits. In: GECCO, pp. 913–920 (2008)
24. Belluz, J., Gaudesi, M., Squillero, G., Tonda, A.: Operator selection using improved dynamic multi-armed bandit. In: GECCO, pp. 1311–1317 (2015)
25. Wang, C., Deng, Y., Li, X., Xin, Y., Gao, C.: A label-based nature heuristic algorithm for dynamic community detection. In: PRICAI, pp. 621–632. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29911-8_48
26. Zhen, H., Gong, W., Wang, L.: Evolutionary sampling agent for expensive problems. IEEE Trans. Evol., Comput. (2022)
27. Olson, R.S., La Cava, W., Orzechowski, P., Urbanowicz, R.J., Moore, J.H.: PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min. **10**(1), 1–13 (2017)
28. Ni, J., Drieberg, R.H., Rockett, P.I.: The use of an analytic quotient operator in genetic programming. IEEE Trans. Evol. Comput. **17**(1), 146–152 (2012)
29. Helmuth, T., McPhee, N.F., Spector, L.: The impact of hyperselection on lexicase selection. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 717–724 (2016)

Evolving a Better Scheduler for Diffusion Models

Zheping Liu[1](✉), Andy Song[1], Nasser Sabar[2], and Wenkai Li[1]

[1] RMIT University, Melbourne, VIC, Australia
s3811732@student.rmit.edu.au
[2] La Trobe University, Melbourne, VIC, Australia

Abstract. AI Generated Content (AIGC) is becoming phenomenally prominent and impactful. One of the key generative algorithms used in AIGC is the diffusion model which is widely used in generative images and audio. In comparison with other generative methods such as GAN (Generative Adversarial Network) and VAE (Variational Auto Encoder), diffusion models can generate samples of higher quality. To further improve diffusion models, especially in terms of sampling speed, we propose an evolutionary algorithm in this paper. That is to enhance the noise scheduler of the diffusion framework, thereby improving both performance and sampling speed. This is the first diffusion model that incorporates evolutionary algorithms. Our experiments show that evolved schedulers can bring concrete improvement in the generative process.

Keywords: Evolutionary Algorithms · Mutation Operators · Diffusion Models · Image Generation

1 Introduction

Generative artificial intelligence (GAI) has become a highly celebratised topic in recent years. Prominent examples include large language models like the generative pre-trained transformer (GPT), which can handle human interaction in natural languages and produce answers that closely resemble human-like responses. Other than text information, generative models can produce images and audio, such as DALL-E, Stable Diffusion, and Midjourney, which can generate incredibly realistic images based on provided keywords or prompts. In many areas e.g. customer services, consultancy, media, and design, generative models have demonstrated an undeniable potential to revolutionize the current practice.

One of the most promonient generative models is the diffusion model. It has rapidly gained increasing popularity among researchers. It also serves as a key algorithm for various generative tools for image and audio data. The Denoising Diffusion Probabilistic Models (DDPM) [7] is often considered as a significant milestone as DDPM brings the performance of diffusion models to the state-of-the-art level. DDPM establishes a fundamental framework for diffusion models,

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 398–409, 2024.
https://doi.org/10.1007/978-981-99-7022-3_37

comprising two main components: the noise scheduler and a U-net neural network model. Unlike the previous best-performing generative algorithm, Generative Adversarial Network (GAN) [5], DDPM stands out in several ways. Firstly, while GAN requires two distinct networks trained in an adversarial or competitive manner, DDPM only requires one network. This approach dramatically simplifies the training process, mitigating issues like the vanishing gradient [1] problem. Secondly, in GAN, the generator transforms noise into desired data, whereas in DDPM, the neural network predicts the noise added during the diffusion process at each time step. Subsequently, a sampling method is employed to eliminate this noise from randomly generated noise data and progressively "recover" it to realistic data. In other words, the sampling process in DDPM necessitates the noise data to undergo a complete backward process, typically involving hundreds or even thousands of steps, depending on the size of the data. Nonetheless, a notable limitation of diffusion models is their slow inference speed. This is a place where diffusion models can be improved.

In this study, a novel method is proposed to enhance the inference performance of diffusion models by utilizing evolutionary algorithms, which are to evolve a more efficient noise scheduler. Two major contributions are as follows:

- Introducing a novel evolutionary component in diffusion model to improve both the inference speed and the quality of outputs;
- Revealing the impact of various key parameters in the noise scheduler on the performance of inference.

2 Related Works

Diffusion models can be divided into two parts, forward process and backward process. Given some real data x_0 from a real distribution p_{real}, and sample some noise z from Gaussian distribution $\mathcal{N}(\mu, \sigma)$, in the forward process, the sampled noise z is gradually added to the real data x_0 over a large number of steps T. The output of the forward process, denoted as x_T, represents noisy data that closely resembles the noise distribution. On the other hand, during the backward process, the diffusion model aims to restore the noisy data x_T back to its original state, x_0. The forward and backward processes can be mathematically represented by Eqs. 1 and 2 respectively.

$$\begin{aligned} q(\mathrm{x}_{1:T}|\mathrm{x}_0) &:= \prod_{t=1}^{T} q(\mathrm{x}_t|\mathrm{x}_{t-1}), \\ q(\mathrm{x}_t|\mathrm{x}_{t-1}) &:= \mathcal{N}(\mathrm{x}_t; \sqrt{1-\beta_t}\mathrm{x}_{t-1}, \beta_t\mathbf{I}) \end{aligned} \tag{1}$$

$$\begin{aligned} p_\theta(\mathrm{x}_{0:T}) &:= p(\mathrm{x}_T) \prod_{t=1}^{T} p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t), \\ p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) &:= \mathcal{N}(\mathrm{x}_{t-1}; \mu_\theta(\mathrm{x}_t, t), \Sigma_\theta(\mathrm{x}_t, t)) \end{aligned} \tag{2}$$

Denoising Diffusion Probabilistic Models (DDPM) [7] is one the earliest diffusion models that achieves state-of-the-art performance in generative area. DDPM designs a neural network (U-Net) to predict the noise that has been added to the real data at each forward step t, where $t \in [1, T]$. The backward process is actually the opposite, where a complete image can be generated out of given Gaussian noise. The noise is gradually removed through step T to 0 by the neural network that is trained to predict noise to be removed at each step. Then the output of the process is a generated sample that is similar to the real sample. In DDPM, the diffusion or forward process relies on a set of pre-defined noise schedules $\{\beta\}_1^T$, and it can be simulated for any step $t \in [1, T]$ with the closed formula in Eq. 3.

$$y_t = \sqrt{\bar{\alpha}_t} y_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon,\ \alpha_t = 1 - \beta_t,\ \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i \tag{3}$$

where y_t is the diffused data at time step t, y_0 is the original data at time step 0; β_t are the pre-defined noise parameter at time step t. Soon after, a study by Song et al. [20] introduced a novel approach for modeling diffusion models. In their work, the diffusion process is treated as a stochastic differential equation (SDE). Instead of predicting the noises used for data diffusion, they train a neural network to estimate the "score" and employ Langevin dynamics to sample the generated data. More importantly, they demonstrated that the Denoising Diffusion Probabilistic Models (DDPM) could be modeled within their framework as well. The forward and backward processes can be represented by the following SDEs, as shown in Eqs. 4 and 5.

$$\mathrm{d}\mathtt{x} = \mathtt{f}(\mathtt{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathtt{w} \tag{4}$$

$$\mathrm{d}\mathtt{x} = [\mathtt{f}(\mathtt{x}, t) - g^2(t)\nabla_{\mathtt{x}} \log p_t(\mathtt{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathtt{w}} \tag{5}$$

Diffusion models have gained significant popularity due to their ability to generate outputs with superior fidelity. In a study by Dhariwal et al. [3], it was reported that diffusion models exhibited higher quality results on various datasets, both in unconditional and conditional settings, when compared to other state-of-the-art image generative algorithms, including GANs. However, one drawback of both the original Denoising Diffusion Probabilistic Models (DDPM) and the SDE diffusion model is that they require a complete inference process with T inference steps to generate a batch of data. In order to ensure that the noise added at each step remains sufficiently small, the value of T is often set to a large number (which may also depend on the size of the data), such as 1000 or even larger. As a consequence, the inference process takes longer compared to other methods like GANs, making it a major limitation of diffusion models.

In recent years, extensive research has been conducted on diffusion models with the aim of addressing the aforementioned limitation and further enhancing their generative performance. Researchers have explored various approaches and techniques to improve the efficiency and effectiveness of diffusion models. These efforts have led to advancements in areas such as accelerated inference methods, optimization algorithms for noise scheduling, architectural modifications to neural networks, and novel training procedures. Song et al. [18] introduce Denoising Diffusion Implicit Models (DDIM), which models the diffusion process as a deterministic process without the addition of random noise. This unique approach allows DDIM to bypass the need for inference through the entire T steps that the model is trained with. In other words, the inference step T' does not have to be the same as the training step T. This flexibility enables the diffusion model to utilize a significantly smaller inference step during the inference phase, resulting in a considerable acceleration of the inference process. However, it is important to note that there exists a trade-off between the speed of inference and the quality of the generated data when using DDIM. Additionally, Nichol et al. [14] propose an improved version of the original Denoising Diffusion Probabilistic Models (DDPM). Their approach incorporates a new hybrid learning objective and utilizes a cosine noise schedule instead of a linear schedule. They claim that their enhanced DDPM can directly reduce the number of required inference steps while only experiencing a negligible reduction in the quality of the generated data. This advancement aims to strike a balance between faster inference and maintaining high-quality outputs. [19] is a follow-up research on score-based diffusion models. They proposed several techniques to improve the generative quality of high-resolution image datasets. These techniques include the following aspects: 1) manipulating the initial noise scale; 2) using a T that is as large as possible; 3) during the sampling phase, applying exponential moving average (EMA) to parameters. [12] proposed a method called PriorGrad that tunes the prior distribution from a simple Gaussian distribution to a data-dependent adaptive distribution.

Apart from reporting that the diffusion model has outperformed GANs, [3] also proposed a method called classifier guidance. This method was inspired by the utilization of class label information in GANs. They trained a classifier using the noisy images at each time step t and utilized the gradients of this trained classifier to steer the outputs towards the desired class label. In their work, [9] further built upon previous research by incorporating a discriminator into their approach. Instead of solely relying on a classifier, they trained a discriminator inspired by GANs. Similar to the classifier mentioned earlier, the discriminator was trained using the noisy images at each time step t. During training, diffused real images from the training dataset were considered as true, while diffused generated images were considered as false. The output of the discriminator was then combined with the computed score in the SDE diffusion model. The authors reported new state-of-the-art results on CIFAR-10 [10] and ImageNet [2] datasets under both conditional and unconditional settings. In a different approach from the methods mentioned above, [16] developed a progressive method based on knowledge distillation [6]. In each iteration, they constructed a student model with half the number of inference steps compared to the previous iteration and distilled knowledge from the previous model (referred to as the teacher model). Through several iterations, they achieved state-of-the-art performance on CIFAR-10 by using only 4 sampling steps. Similarly, [13] also employed knowledge distillation to reduce the inference steps. However, unlike the previous method, they established a training objective that minimizes the difference between the generated distributions of the student and teacher models. Additionally, their method trains the student model only once instead of progressively reducing the sampling steps.

In addition to the methods that focus on improving neural networks in diffusion models, there are also approaches that work on the noise schedule. [17] introduced a neural network, denoted as P_θ, which can estimate a more optimal noise parameter $\sqrt{\bar{\alpha}_t}$ at any given time step t. The idea behind this approach is to provide continuous control and adjustment for the noise level between different steps, allowing for better performance. Similarly, the bilateral denoising diffusion model (BDDM) [11] [8] trains a neural network to generate a surrogate noise schedule that replaces the original one during the inference phase. One advantage of BDDM compared to others is that the inference steps in this new noise schedule can be reduced, resulting in a significantly higher inference speed.

GAN, as a prominent generative algorithm, has been actively studied and integrated into diffusion models. In work [22] Xiao et al. argued that the requirement for large sampling steps in diffusion models stemmed from the Gaussian assumption during the denoising phase. To address this, they proposed the denoising diffusion GAN, which utilizes a multimodal conditional GAN to re-model the diffusion distribution. On a similar note, [21] incorporated the forward diffusion process into a GAN framework. The discriminator in their approach, similar to [9], classifies diffused real data and generated data. The authors demonstrated that the generator can effectively learn from the discriminator feedback, leveraging the diffusion process, as supported by both theoretical and experimental evidence. Along a similar line, [23] introduced a method with a similar idea but also incorporated a classifier to provide further guidance in the diffusion process. Furthermore, [4] aimed to unify GAN and score-based diffusion models by integrating the generator component into the diffusion model. This unification allows for leveraging the strengths of both approaches.

3 Methodology

The proposed method in this study integrates an evolutionary algorithm to search for a better noise scheduler, also known as a scheduler, in diffusion models like DDPM and DDIM. The ordinary definition of a DDPM/DDIM noise scheduler is as follows:

$$Noise\ Scheduler := \mathcal{F}(\beta_{start},\ \beta_{end},\ steps_{train},\ steps_{inference},\ schedule\ method) \tag{6}$$

In DDPM/DDIM noise schedulers, a list of betas, denoted as $\beta_{i}{}_{1}^{T}$, is generated. Here, T represents the number of training/inference steps. The values β_{start} and β_{end} determine the starting value, β_0, and the end value, β_T, respectively. The schedule method refers to a mapping that converts a range of betas into a sequence of betas. The schedule method can be linear, scaled linear, or squared cosine. This list of βs determines the amount of noise that is added to the

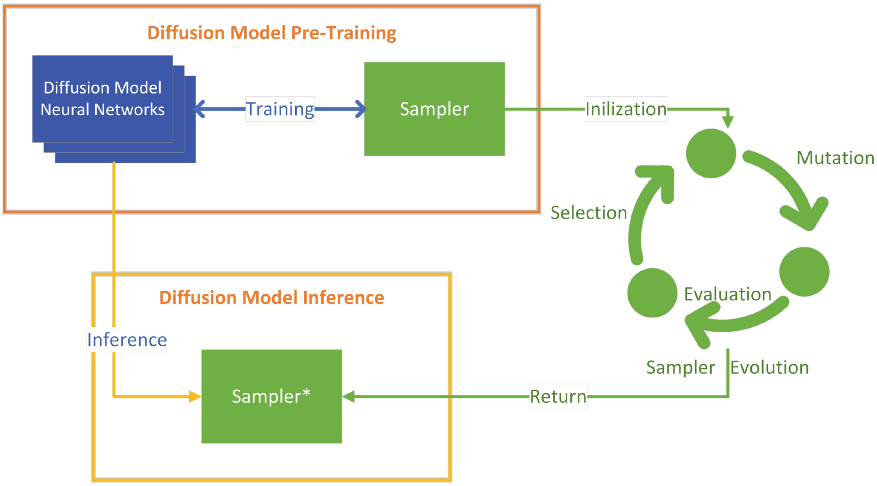


Fig. 1. Flow of Evolutionary Scheduler. The whole algorithm can be decomposed into three steps. 1) pre-training the diffusion network using the original scheduler; 2) evolving the scheduler using the proposed evolutionary algorithm; 3) inference images using diffusion network trained from Step 1 and scheduler evolved from Step 2.

images during the forward process or removed from the noisy images during the backward or inference process at specific time steps.

In most diffusion model research, the noise scheduler is typically pre-defined before training or fixed during training as a hyperparameter. The noise scheduler plays a crucial role in achieving better performance, and thus, it needs to be carefully tuned. Evolutionary algorithms have been commonly used to optimize hyperparameters in machine learning methods. This inspiration led us to propose our method, the evolutionary noise scheduler in diffusion models.

Originally, our method involved evolving the list of betas (${\beta_i}_1^T$) directly, which is generated by the noise scheduler. This approach aimed to optimize the performance by finding the best betas. However, the step size T is often a large number, such as 1000, resulting in each gene in the population containing a large number of individual parts to be evolved. The computational resources required to evolve such a population become prohibitively expensive. On the other hand, the noise scheduler defined in Eq. 6 has only five different parameters. By modifying these parameters, it is possible to generate different betas.

The flow of our proposed method is shown in Fig. 1. The design of the evolutionary algorithm strictly follows the four steps of evolutionary search. The algorithm pseudo-code is displayed in Algorithm 1 and explained below.

Initialization. N noise schedulers are initialized. Based on empirical prior studies, it is beneficial to initialize one of the schedulers using the original parameters that were used during training. The remaining schedulers are randomly initialized by sampling values from a Gaussian distribution, specifically $\mathcal{N}(\mu, \sigma)$, for both β_{start} and β_{end}. The mean values μ for each parameter are set to the corre-

Algorithm 1. Evolutionary Scheduler Algorithm

```
Require: mutation size $m$; population size $n = 1$; original scheduler $S$; expected value
  of $\beta_{start}$, $\mu_{start} = 0.0001$; expected value of $\beta_{end}$, $\mu_{end} = 0.02$; pre-defined variance
  of $\beta_{start}$, $\sigma_{start} = 0.00005$; pre-defined variance of $\beta_{end}$, $\sigma_{end} = 0.005$; mutation
  step size $s = 0.01$;
  Initialization
  $S_1 = S(\mu_{start},\ \mu_{end},\ step_{train},\ step_{inference})$
  for $i = 2, .., n$ do
    sample $\beta_{start}$ from $\mathcal{N}(\mu_{start}, \sigma_{start})$
    sample $\beta_{end}$ from $\mathcal{N}(\mu_{end}, \sigma_{end})$
    $S_i = S(\beta_{start},\ \beta_{end},\ step_{train},\ step_{inference})$
  end for
  $\{S\} = \{S_1,\ \dots\ , S_n\}$
  for Training Epochs do
    Variation
    for $S_i$ in $\{S\}$ do
      for mut in $\{mutation\}$ do
        $S'_i = \mathcal{M}(S_i,\ mut)$
      end for
    end for
    $\{S'\} = \{S'_1,\ ...,\ S'_{n \times m}\}$
    Evaluation
    for $S'_i$ in $\{S'\}$ do
      $fitness_i = \mathcal{F}(S'_i)$
    end for
    fitness_scores = $\{fitness_1,\ ...,\ fitness_{n \times m}\}$
    Selection
    fitness_scores = argsort(fitness_scores)
    $\{S\} = \{S_{fitness\_scores_0},\ ...,\ S_{fitness\_scores_{n-1}}\}$
  end for
  return $\{S\}$
```

sponding original scheduler values, while the variances σ are manually specified. The training steps, inference steps, and schedule method of the initialized schedulers will remain the same as those of the original scheduler.

Variation. During the variation phase of the algorithm, we solely use the mutation operators. The following mutations are defined: 1) increasing β_{start}, 2) decreasing β_{start}, 3) increasing β_{end}, 4) decreasing β_{end}, 5) increasing $steps_{inference}$ and 6) decreasing $steps_{inference}$. Note that mutations (5) and (6) only apply when dealing with DDIM noise schedulers since they do not require the inference steps to be equal to the training steps. In contrast, for DDPM noise schedulers, the training steps and inference steps must be the same. Altering the training steps is not included in our mutation operators due to the poor performance observed, as discussed in later sections. The step sizes for betas and inference steps are denoted as ϵ and δ, respectively. The variances of β_{start} and β_{end} are represented as σ_{start} and σ_{end}.

Evaluation. The fitness evaluation stage in our proposed method is crucial. We designed three different types of fitness calculations, including comparing generated images, comparing added noises and utilizing guidance from the discriminator. Prior studies show that utilizing guidance from the discriminator is the current best. The performance of the other fitness calculation methods is discussed in Sect. 5. Discriminator guidance was originally proposed in [9]. They use the outputs of the discriminator to improve the score in score-based diffusion models. The discriminator in our method takes the time step t and the given images at the t-th step as inputs. It then outputs the probabilities, indicating whether these images are blurred versions of real images from the provided dataset or generated images. A higher probability indicates a better diffused effect. This characteristic of the discriminator makes it an ideal evaluator for assessing the performance of different noise schedulers. The detailed evaluation process is as the following. Firstly, sample a batch of time steps $\{t\}$ from range $[1, T]$, and a batch of real images $\{x_0\}$ from the training data set. Secondly, for every offspring generated from the previous variation step, add noise to $\{x_0\}$ with $\{t\}$, and get noisy images $\{x_t\}$. Third and finally, input $\{x_t\}$ and $\{t\}$ to the discriminator to get the probabilities of them being real noisy images, then, calculate their average values and output those as the fitness score for each noise scheduler.

Algorithm 2. Fitness Evaluation

```
Require: batch size $b$; discriminator $\mathcal{D}$; maximum inference step $T$;
  Sample a batch of time steps $\{t_i\}_1^b$, where $t_i \in [1, T]$
  Sample a batch of real images $\{x_0^i\}_1^b$
  for $S_i$ in $S$ do
    $\{x_t^i\} = \text{AddNoise}\ (S_i, \{x_0^i\}_1^b, \{t_i\}_1^b)$
    $\text{fitness\_score} = \frac{1}{b}\sum_{i=1}^{b}(\mathcal{D}(x_t^i, t_i))$
  end for
  $\text{fitness\_scores} = \{\text{fitness\_score}_i\}_{i=1}^{n\times m}$
  return fitness_scores
```

Selection. The selection process uses a straightforward method, which simply picks the top-n offspring with the highest fitness scores to proceed to the next iteration.

4 Experiments

Our experiments are implemented based on the open package *diffusers* [15]. The dataset is the well-known CIFAR-10. The Frechet Inception Distance (FID) is computed as the evaluation metric for assessing the quality of the generated samples. A lower FID indicates better quality. The original diffusion model is trained using 500 training and inference steps. All experiments were conducted on an NVIDIA A100 GPU. DDIM results with varying numbers of inference steps are used as benchmarks in our experiments. This allows us to compare and demonstrate the improvements in FID under the same inference times in each setting. Based on the experimental results shown in Table 1, we observe an increase in performance in almost all settings, with relatively low variance, where the number of inference steps is less than 500 inference steps. In the case of 500 inference steps, our method did not achieve a better result compared to its original scheduler. This could be due to the network being trained and optimized specifically for this noise scheduler with the same inference step, making any modifications to β_{start} and β_{end} ineffective as to the original configuration.

Table 1. Experiment Results

Inference Steps	Benchmark	Our Method	Inference Time(s)
50	19.06	**17.95 ± 0.13**	1675
100	18.48	**17.60 ± 0.43**	3343
150	20.74	**18.53 ± 0.29**	4956
200	29.29	**21.59 ± 0.31**	6616
250	18.16	**17.14 ± 0.23**	8265
500	16.61	17.01 ± 0.19	16448

Furthermore, an interesting observation from our experiments is that the performance of inference steps at 50, 100, and 250 is better than that of 150 and 200. This observation holds true for both the benchmark (DDIM) and our proposed method. Theoretically, one might expect that higher inference steps, approaching the original training steps (i.e., 500), would produce higher quality generated data. However, our experimental results do not support this assumption. Instead, we hypothesize that inference steps that are exactly divisible by the original training steps tend to perform better than other values. This observation suggests that there might be a certain relationship between the number of training steps and inference steps in diffusion models. It highlights the importance of carefully considering the choice of inference steps to achieve the best performance in terms of image quality. Our further research will aim to fully understand and possibly leverage this phenomenon.

5 Discussions

5.1 Optimal β_{start} and β_{end}

Our experiments observe the impact of the two most important parameters, β_{start} and β_{end}, in the noise scheduler. Table 2 shows some representative checkpoints selected in our investigation with the corresponding FID value computed from the experiments where the inference step equals 50. What can be observed from the table is that the β_{end} values of the top-performing offspring are very similar, with slight reductions or almost equal to their starting point of 0.02. Conversely, offspring with higher β_{end} values exhibit poorer FID scores in comparison. As for β_{start}, it is evident that the best-performing offspring have β_{start}

values around 0.00005 or 0.00006, which is approximately half of the original value (i.e., 0.0001). Offspring with values lower than this threshold or higher β_{start} values tend to perform relatively poorly in our experiments. In some rare cases, we encountered negative β_{start} values, which resulted in destructive outcomes. This occurs because negative β_{start} implies the addition of noise to the data instead of its removal in the final stages of the backward process, hence generating worse output.

Table 2. FID and their β_{start} and β_{end}

β_{start}	5.76e-05	6.38e-05	6.33e-05	0.0001295	0.0001588	2.53e-05	0.0001231	7.56e-05	-1.87e-05
β_{end}	0.01988	0.01996	0.02006	0.01917	0.01922	0.02122	0.02048	0.01886	0.02060
FID	17.72	17.96	17.99	19.17	20.17	21.09	21.20	22.22	679

5.2 Choices of Fitness Functions

As described in the methodology, Sect. 3, three different fitness functions were developed in this study. This subsection discusses their advantages and disadvantages respectively (Table 3).

Table 3. Results of Different Fitness Functions

Fitness	Image Comparison	Noise Comparison	Discriminator
FID	17.37	23.76	**17.01**

Image Comparison. This is an intuitive fitness function, as it involves comparing real images x_0 with samples obtained by applying the evolved noise scheduler to diffuse them to a given time step t ($t \in [1, T']$). The trained diffusion model and the same evolved noise scheduler are then used to backward-diffuse x_t to obtain x'_0. This fitness score is calculated based on the L1 or L2 distances between x_0 and x'_0. This fitness function performs quite well in terms of its outputs. It guides the offspring to evolve in the correct direction, and the achieved FID is very close to the intermediate results of our final method. However, the main limitation of this method is the inference speed. Sampling a batch of images from x_t to x'_0 during each evaluation step can be very time-consuming, particularly when dealing with large values of t. We attempted to limit the sampled t to a smaller range, such as 50. However, even on our experimental machine, training just 5 iterations could take more than 24 h.

Noise Comparison. Since sampling diffused images back to their original forms is time-consuming, we proposed another fitness function that directly compares the added noise and the predicted noise. Similar to the previous method, we sample a batch of real images x_0 and a time step t, and then diffuse the images to x_t. Next, we sample another batch of random noise z from a Gaussian distribution and add it to x_t in just one step, resulting in diffused images x_{t+1}. We input x_{t+1} and $t+1$ to the trained diffusion neural network to predict the added noise z'. Finally, we compute the fitness score using the L1 or L2 distance between z and z'. Unlike the previous method that predicts all noises from $t+1$ to 0, the *noise comparison* fitness function only predicts the noise once for each individual image. This significantly reduces the computational requirements. However, based on our experiments, this fitness function does not provide effective guidance to facilitate evolution. Offspring does not seem to converge to a better state in repeated runs.

Discriminator Fitness. The details of the discriminator are described in Sect. 3. Besides its fabulous performance, discriminator is also highly time-efficient and is not affected by the number of sampled inference steps t.

5.3 Population Size

The population size is a typical hyper-parameter in evolutionary algorithms. In most of our settings, the population size is set to 1, to restrain the computational cost. We have also tested larger population sizes, such as 2 and 3. However, despite the dramatic increase in training time, the performance improvement is negligible. Upon close analysis of the intermediate training status, we observed that the remaining offspring in the same iteration were very similar to each other. The randomly generated initial offspring were usually quickly discarded since they couldn't outperform the original offspring.

6 Conclusion and Future Work

This study presented a novel algorithm that incorporates evolutionary algorithms into diffusion models to search. This approach has allowed us to discover improved noise schedulers, resulting in improvements in both performance and efficiency. By reducing the number of inference steps required, our method offers a more efficient generative process while maintaining high-quality outputs. Overall, our proposed evolutionary diffusion model contributes to advancing diffusion models and opens up new possibilities for their applications in real-world practices.

This study leads to a series of future work. Firstly, one of the key parameters in the noise scheduler, the schedule method, can be further studied. Currently, there are three different schedule methods: linear, scaled linear, and squared cosine. We plan to incorporate mutation of the schedule method to find the optimal one based on the circumstance. Secondly, our observations show that the output score of the discriminator is related to the inputted time step. Through further investigation of the discriminator, we may obtain better guidance from the discriminator by identifying a more accurate time step.

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. Stat 1050, 17 (2017)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
3. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 8780–8794 (2021)
4. Franceschi, J.Y., et al.: Unifying GANs and score-based diffusion as generative particle models. arXiv preprint arXiv:2305.16150 (2023)
5. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27 (2014)
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. Stat 1050, 9 (2015)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
8. Huang, R., et al.: FastDiff: a fast conditional diffusion model for high-quality speech synthesis. In: IJCAI (2022)
9. Kim, D., Kim, Y., Kang, W., Moon, I.C.: Refining generative process with discriminator guidance in score-based diffusion models. Comput. Vis. Pattern Recogn. (2022)
10. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
11. Lam, M.W., Wang, J., Su, D., Yu, D.: BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis. In: ICLR (2022)
12. Lee, S.G., et al.: PriorGrad: Improving conditional denoising diffusion models with data-driven adaptive prior. In: ICLR (2021)
13. Luhman, E., Luhman, T.: Knowledge distillation in iterative generative models for improved sampling speed. arXiv preprint arXiv:2101.02388 (2021)
14. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning, pp. 8162–8171. PMLR (2021)
15. von Platen, P., et al.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers (2022)
16. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022)
17. San-Roman, R., Nachmani, E., Wolf, L.: Noise estimation for generative diffusion models. Comput. Vis. Pattern Recogn. (2021)
18. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
19. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. Adv. Neural. Inf. Process. Syst. **33**, 12438–12448 (2020)
20. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
21. Wang, Z., Zheng, H., He, P., Chen, W., Zhou, M.: Diffusion-GAN: Training GANs with diffusion. arXiv preprint arXiv:2206.02262 (2022)
22. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion GANs. In: ICLR (2022)
23. Yeom, T., Lee, M.: DuDGAN: improving class-conditional GANs via dual-diffusion. arXiv preprint arXiv:2305.14849 (2023)

Investigating the Existence of Holey Latin Squares via Satisfiability Testing

Minghao Liu[1,5], Rui Han[1,5], Fuqi Jia[1,5], Pei Huang[3], Feifei Ma[1,2,5](✉),
Hantao Zhang[4], and Jian Zhang[1,5](✉)

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
{liumh,maff,zj}@ios.ac.cn
[2] Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[3] Stanford University, Stanford, CA, USA
[4] Computer Science Department, The University of Iowa, Iowa City, IA 52242, USA
[5] University of Chinese Academy of Sciences, Beijing, China

Abstract. Holey Latin square (HLS) is a special combinatorial design of interest to mathematicians and is helpful in the construction of many important structures in design theory. In this paper, we investigate the existence of HLSs satisfying the seven kinds of identities with automated reasoning techniques. We formulate this problem as propositional logic formulae. Since state-of-the-art SAT solvers have difficulty solving many HLS problems, we further propose a symmetry breaking method, called partially ordered HLS (POHLS), to eliminate isomorphic solutions. We have achieved the following goals through experimental evaluation. First, we have solved a dozen of open problems interested by mathematicians. Second, we identify the impact of different encodings. Third, we demonstrate the advantages of SAT solver over other FOL-based solvers. Fourth, we show that the proposed POHLS reduction can improve the efficiency of solving and find the complementarity between two types of symmetry breaking techniques.

Keywords: Holey Latin square · Combinatorial designs · Symmetry breaking

1 Introduction

Automated reasoning is one of the key components of artificial intelligence. In recent decades, a series of great progress has been made to solve hard problems in combinatorics by modeling them as logical formulae and solving via automated reasoning techniques. This powerful paradigm has attracted the attention of both mathematicians and computer scientists. Schur number five, a century-old problem, was successfully resolved recently, using SAT solvers [15]. Moreover,

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 410–422, 2024.
https://doi.org/10.1007/978-981-99-7022-3_38

in 2021 automated reasoning solvers produced nonexistence certificates of the Lam's problem [5]. Among various hard problems in combinatorics, we are interested in investigating the existence of *Latin squares* with some specific properties. A Latin square of order n is an $n \times n$ matrix filled by the elements from a set $\{1, 2, \ldots, n\}$, such that an element appears exactly once in each row and each column. The popular Sudoku can also be seen as a special variant of Latin square. Research on this important combinatorial structure can be traced back to Leonhard Euler, who proposed a famous conjecture about the existence of mutually orthogonal Latin squares (MOLS) that was disproved by computer search in 1963 [26]. A number of important applications are related to Latin squares, such as experimental designs [14], error correcting codes [8] and cryptography [25]. Completing partial Latin squares has been proved to be NP-complete [6]. Initially, the existence of Latin squares was mainly found by mathematicians through manual construction. However, the recent developments are strongly supported by automated reasoning. Since the 1990s, the existence of Latin squares of small orders are studied using finite model generators such as *MGTP* [12], *SEM* [30], and SAT solvers such as *SATO* [28], *DDPP* [27], respectively. In recent years, people have proposed different techniques to improve the efficiency of reasoning, and more open problems about Latin squares are resolved [17,19].

In this paper, we investigate the existence of *holey Latin squares* (HLSs). The use of holes, as a carefully designed relaxation of the original structures, is one of the most powerful tools in combinatorial design theory, which has helped mathematicians find the existence of many advanced structures such as Steiner pentagon system [20] and t pairwise orthogonal diagonal Latin squares [1]. The formal definition of HLS is deferred to Sect. 2. According to the definition, some cells in HLS should be empty due to the existence of holes, while necessary properties must still be valid in the hole-free area. More specifically, we mainly focus on finding the existence of HLSs satisfying the seven specific properties (called identities) summarized by mathematicians [3]. Technically, we encode the existence problem of HLS as propositional logic formulae, considering the superiority of SAT solving techniques for many difficult problems [16,18,31]. Our encoding is highly universal, meaning that it can be easily applied to HLS instances with different orders, hole types and identities. Although some simple cases can be solved in the original encoding, the time consumption is not satisfactory when the order of HLS becomes larger. In order to eliminate isomorphic solutions and reduce the search space, we propose a symmetry breaking method called partially ordered HLS (POHLS), in which the priorities of some elements in the first row are fixed. We prove that the existence of any HLS and its corresponding POHLS are equivalent.

We work on a benchmark of 273 HLS instances, which includes a number of open cases and is provided by Lie Zhu, a renowned mathematician. Three ways to improve the solving efficiency are evaluated. First, at-most-one (AMO) is one of the most important constraints for this problem. We compared three kinds of AMO encodings and find that suitable encoding can significantly improve the performance of SAT solvers. Second, we wonder whether other techniques based on first-order logic (FOL) are more suitable for solving this problem. Compared to three powerful FOL-based solvers, the SAT solver shows the best performance on the benchmark. Third, we apply two kinds of techniques: the syntax-based automatic symmetry breaking tools, as well as the proposed semantic-based POHLS reduction. Experiments show that symmetry breaking can further accelerate the resolution of instances, and these techniques can be complementary to each other. As a result of our efforts in technical improvements, 222 new existence results are determined within a one-week time limit per instance. Appendix, source code and benchmarks can be found at: https://github.com/minghao-liu/HLS.

2 Preliminaries

A *quasigroup* is an algebraic structure $(Q, *)$, where Q is a non-empty set and $*$ is a binary operation, satisfying the property that for every pair of elements $a, b \in Q$, the equations $a * x = b$ and $y * a = b$ are uniquely solvable for variables $x, y \in Q$. A *Latin square* is known as the multiplication table of a quasigroup, which can be defined as follows.

Definition 1 (Latin Square). *Given a non-empty set $Q = \{1, 2, \ldots, n\}$, a Latin square L of order n is an $n \times n$ matrix filled by the elements of Q, and every element occurs exactly once in each row and each column.*

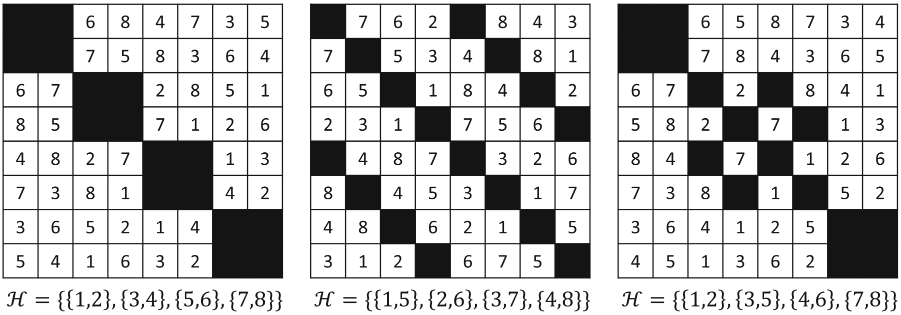
In addition to the simplest form of Latin squares, researchers have long been interested in the existence of Latin squares with some specific properties. This is because many of them have a close relationship to the construction of advanced combinatorial structures, such as orthogonal array (OA) and pairwise balanced design (PBD) [7]. These properties of Latin squares are commonly represented as equations, which are called short conjugate-orthogonal identities (abbreviated as identity). In 1975, Evans [10] summarized the non-trivial identities systematically, and Bennett [3] further simplified the number to seven. These identities and their common names (if any) are listed as follows:

(1)	$(x * y) * (y * x) = x$	Schröder's second law; Schröder quasigroup
(2)	$(y * x) * (x * y) = x$	Stein's third law
(3)	$((x * y) * y) * y = x$	C_3-quasigroup
(4)	$x * (x * y) = y * x$	Stein's first law; Stein quasigroup
(5)	$((y * x) * y) * y = x$	
(6)	$(y * x) * y = x * (y * x)$	Stein's second law
(7)	$(x * y) * y = x * (x * y)$	Schröder's first law

Furthermore, the definition of holey Latin square is shown as follows:

Definition 2 (Holey Latin Square). *Given a non-empty set $Q = \{1, 2, \ldots, n\}$ and a hole set $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$ such that $H_i \subseteq Q$ for $1 \leq i \leq m$, $H_1 \cup H_2 \cup \cdots \cup H_m = Q$ and $H_i \cap H_j = \emptyset$ for $1 \leq i, j \leq m$ $(i \neq j)$, a holey Latin square L is an $n \times n$ matrix satisfying the following properties:*

(1) Every cell of L is either filled by an element of Q or empty;
(2) Every element occurs at most once in each row and each column;
(3) A cell $L(x, y)$ is empty if and only if there exists a hole $H_i \in \mathcal{H}$, such that $x \in H_i$ and $y \in H_i$;
(4) An element $x \in Q$ occurs in the y-th row/column if and only if there does not exist a hole $H_i \in \mathcal{H}$, such that $x \in H_i$ and $y \in H_i$.



■	■	6	8	4	7	3	5
■	■	7	5	8	3	6	4
6	7	■	■	2	8	5	1
8	5	■	■	7	1	2	6
4	8	2	7	■	■	1	3
7	3	8	1	■	■	4	2
3	6	5	2	1	4	■	■
5	4	1	6	3	2	■	■

$\mathcal{H} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$

■	7	6	2	■	8	4	3
7	■	5	3	4	■	8	1
6	5	■	1	8	4	■	2
2	3	1	■	7	5	6	■
■	4	8	7	■	3	2	6
8	■	4	5	3	■	1	7
4	8	■	6	2	1	■	5
3	1	2	■	6	7	5	■

$\mathcal{H} = \{\{1,5\},\{2,6\},\{3,7\},\{4,8\}\}$

■	■	6	5	8	7	3	4
■	■	7	8	4	3	6	5
6	7	■	2	■	8	4	1
5	8	2	■	7	■	1	3
8	4	■	7	■	1	2	6
7	3	8	■	1	■	5	2
3	6	4	1	2	5	■	■
4	5	1	3	6	2	■	■

$\mathcal{H} = \{\{1,2\},\{3,5\},\{4,6\},\{7,8\}\}$

Fig. 1. An example of three isomorphic $\text{HLS}^{(5)}(2^4)$ with different hole sets $\mathcal{H}$.

The *type* of a hole set $\mathcal{H}$ is typically represented as a string in the form ${h_1}^{m_1}{h_2}^{m_2}\ldots{h_r}^{m_r}$ ($h_1 < h_2 < \cdots < h_r$), which denotes the presence of m_1 holes of size h_1, m_2 holes of size h_2, and so on. For example, given that $\mathcal{H} = \{\{1, 3\}, \{2, 6\}, \{4\}, \{5, 7\}\}$, its type should be denoted as $1^1 2^3$.

Moreover, if for any $x, y \in Q$ such that $L(x, y)$ is non-empty, the computation process of identity (i) based on x, y does not reference any empty cells and the identity is holds, we can say that HLS L satisfies identity (i). If a holey Latin square with hole type $\mathcal{T}$ satisfies identity (i), it can be denoted as $\text{HLS}^{(i)}(\mathcal{T})$. Nevertheless, HLSs that satisfy one of the seven identities mentioned above have yet to be systematically investigated. Our work, aiming to find the existence of HLSs satisfying specific identities through SAT solving, yields a number of new findings presented in Sect. 5.

3 Modeling

In this section, we introduce the methodology to model the existence problem of HLS. We establish a standardized form, which facilitates the encoding into propositional logic formulae in a universal manner.

3.1 Standardization

In the previous notation, the introduction of hole type $\mathcal{T}$ is because it can serve as a standardized representation for a group of isomorphic HLSs. There is an important property about hole type and its corresponding hole sets:

Proposition 1. *If there exists an* $\mathrm{HLS}^{(i)}(\mathcal{T})$ L_1 *with hole set* $\mathcal{H}_1 = \{H_1, H_2, H_3, \ldots, H_m\}$*, then there exists another* $\mathrm{HLS}^{(i)}(\mathcal{T})$ L_2 *with hole set* $\mathcal{H}_2 = \{H_1', H_2', H_3, \ldots, H_m\}$*, such that* $a \in H_1$*,* $b \in H_2$*,* $H_1' = (H_1 \backslash \{a\}) \cup \{b\}$*, and* $H_2' = (H_2 \backslash \{b\}) \cup \{a\}$*.*

The proof can be found in the Appendix. From Proposition 1, it is equivalent to asserting the existence of two HLSs with the same hole type, as one can be transformed into the other within a finite number of steps. Figure 1 is an example that demonstrates the mutual convertibility of HLSs with different hole sets. In the modeling, to rebuild the hole set $\mathcal{H}$ from $\mathcal{T}$, we have an ordered assignment of the elements in each hole. For example, given $\mathcal{T} = 1^3 2^2$, the hole set should be $\mathcal{H} = \{\{1\}, \{2\}, \{3\}, \{4, 5\}, \{6, 7\}\}$. For convenience, we set up a Boolean constant matrix C of order n, such that for any $x, y \in Q$, $C_{x,y} = 1$ if and only if $\exists H_i \in \mathcal{H}$, $x \in H_i \wedge y \in H_i$. This matrix encodes the hole set $\mathcal{H}$ as a binary relation on Q.

3.2 SAT Encoding

Next, we demonstrate the method we used for encoding various constraints of $\mathrm{HLS}^{(i)}(\mathcal{T})$ instances into propositional logic formulae. This method is universal, meaning that it can be applied to any possible order, hole type, and identity.

Assuming the HLS we are searching for is denoted by L. First, we define the Boolean variable $l_{x,y,v}$, which means whether $L(x, y) = v$ is true or not. A set of unit clauses is set to forbid illegal assignments due to the presence of holes:

$$\forall x, y, v \in Q, \quad (C_{x,y} \vee C_{x,v} \vee C_{y,v}) \rightarrow \neg l_{x,y,v}. \tag{1}$$

Next, two sets of clauses are introduced to keep the consistency of domains:

$$\begin{aligned} \forall x, y \in Q, &\quad \neg C_{x,y} \rightarrow \textstyle\bigvee_{v \in Q} l_{x,y,v}, \\ \forall x, y \in Q, &\quad AMO\left(\{l_{x,y,v} \mid v \in Q\}\right), \end{aligned} \tag{2}$$

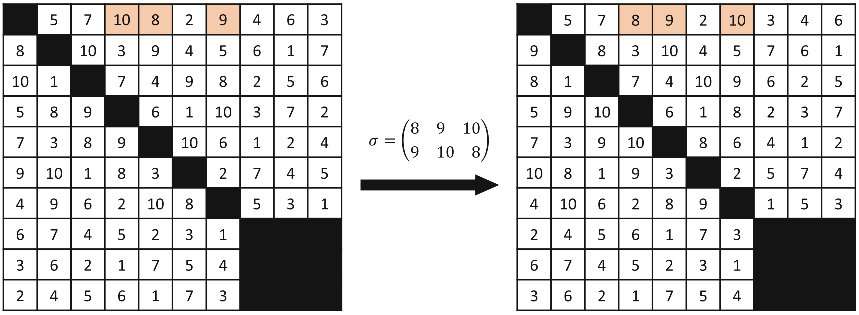
where AMO stands for the “at-most-one” encoding, which means that at most one literal in the set could be true.

The following step is to encode the Latin square property, which means that an element cannot occur twice in the same row or column:

$$\begin{aligned} \forall x, v \in Q, &\quad AMO\left(\{l_{x,y,v} \mid y \in Q\}\right), \\ \forall y, v \in Q, &\quad AMO\left(\{l_{x,y,v} \mid x \in Q\}\right). \end{aligned} \tag{3}$$

Finally, the seven identities can be encoded as follows. Note that for any $v \in Q$, we have assigned $l_{x,y,v}$ to False when $C_{x,y} = 1$, so there is no need to explicitly restrict that the computation step would not go into the holes:

$$\begin{aligned}
&\text{Identity (1)}: &&\forall x,y,u,v \in Q, &&(l_{x,y,u} \wedge l_{y,x,v}) \rightarrow l_{u,v,x},\\
&\text{Identity (2)}: &&\forall x,y,u,v \in Q, &&(l_{y,x,u} \wedge l_{x,y,v}) \rightarrow l_{u,v,x},\\
&\text{Identity (3)}: &&\forall x,y,u,v \in Q, &&(l_{x,y,u} \wedge l_{u,y,v}) \rightarrow l_{v,y,x},\\
&\text{Identity (4)}: &&\forall x,y,u,v \in Q, &&(l_{x,y,u} \wedge l_{x,u,v}) \rightarrow l_{y,x,v},\\
&\text{Identity (5)}: &&\forall x,y,u,v \in Q, &&(l_{y,x,u} \wedge l_{u,y,v}) \rightarrow l_{v,y,x},\\
&\text{Identity (6)}: &&\forall x,y,u,v \in Q, &&(l_{y,x,u} \wedge l_{u,y,v}) \rightarrow l_{x,u,v},\\
&\text{Identity (7)}: &&\forall x,y,u,v \in Q, &&(l_{x,y,u} \wedge l_{u,y,v}) \rightarrow l_{x,u,v}.
\end{aligned} \tag{4}$$



■	5	7	10	8	2	9	4	6	3
8	■	10	3	9	4	5	6	1	7
10	1	■	7	4	9	8	2	5	6
5	8	9	■	6	1	10	3	7	2
7	3	8	9	■	10	6	1	2	4
9	10	1	8	3	■	2	7	4	5
4	9	6	2	10	8	■	5	3	1
6	7	4	5	2	3	1	■	■	■
3	6	2	1	7	5	4	■	■	■
2	4	5	6	1	7	3	■	■	■

a) $HLS^{(1)}(1^7 3^1)$

$\sigma = \begin{pmatrix} 8 & 9 & 10 \\ 9 & 10 & 8 \end{pmatrix}$

■	5	7	8	9	2	10	3	4	6
9	■	8	3	10	4	5	7	6	1
8	1	■	7	4	10	9	6	2	5
5	9	10	■	6	1	8	2	3	7
7	3	9	10	■	8	6	4	1	2
10	8	1	9	3	■	2	5	7	4
4	10	6	2	8	9	■	1	5	3
2	4	5	6	1	7	3	■	■	■
6	7	4	5	2	3	1	■	■	■
3	6	2	1	7	5	4	■	■	■

b) $POHLS^{(1)}(1^7 3^1)$

Fig. 2. An example of an HLS and its corresponding POHLS. In the first row of POHLS, the elements $\{8, 9, 10\}$ in the same hole should occur in ascending order.

4 Symmetry Breaking

Isomorphic solutions are common in many combinatorial problems due to the presence of *symmetry*, and they are also helpful in improving the solvability [17,21,29]. In this section, a static method is proposed to break the symmetries by appending constraints to the formula. Firstly, the partially ordered HLS (POHLS) is defined below, which is closely related to our symmetry breaking method.

Definition 3 (Partially Ordered HLS, POHLS). *A partially ordered HLS L' is a holey Latin square, such that there exists a hole $H_i \in \mathcal{H}$ with $|H_i| \geq 2$, where for every two elements $x, y \in H_i$, if $x < y$ and $L'(1, a) = x$, $L'(1, b) = y$, it must hold that $a < b$.*

Figure 2 shows an HLS instance and its corresponding POHLS. Compared with general HLSs, the order of some elements occurring in the first row of POHLS is constrained. Next, we prove the existence of POHLS.

Proposition 2. *If there exists an* $\mathrm{HLS}^{(i)}(\mathcal{T})$ *denoted as* L*, then there must be a POHLS* L' *also satisfying identity* (i) *with hole type* $\mathcal{T}$*.*

The proof can be found in the Appendix. Proposition 2 suggests that we can search for the existence of POHLS instead of finding HLS directly, which is referred to as *POHLS reduction*. Generally, we choose the largest hole H_m as the ordered one. To break symmetry, the corresponding constraints can be added:

$$\forall x, y \in H_m \ (x < y), \ \forall a \in Q, \quad l_{1,a,y} \rightarrow \bigvee_{b=1}^{a-1} l_{1,b,x}, \tag{5}$$

The clauses generated from Eq. 5 can eliminate $|H_m|!$ isomorphic HLSs theoretically. Meanwhile, our method can also be combined with automatic symmetry breaking tools to achieve better performance. We will further provide more detailed comparison results of SAT solvers under different techniques in Sect. 6.

5 Benchmarks and Results

To investigate the existence of HLSs, we model a group of interesting and challenging benchmarks of orders from 7 to 14, with different identities and hole types. The benchmarks are kindly provided by Lie Zhu[1]. For each case, we encode HLS satisfying the seven identities, respectively, so there are $39 \times 7 = 273$ instances in total, and the existence of most of them remain open.

We present a comprehensive compilation of the existence results of these HLS benchmarks in Table 1, which are determined by running *Kissat* [4], a state-of-the-art SAT solver. For each instance, it runs for a maximum of one week (168 h). Among the 262 existence results, 40 are previously known, so the left 222 are newly reported results. This indicates that our approach can efficiently find HLS instances with diverse structures in a universal manner. For most HLS instances with orders 12–14, symmetry breaking techniques can help reduce their solving time, especially for those unsatisfiable ones. In fact, there are 9 instances that could only be solved with symmetry breaking in our experiments. Besides, as demonstrated later, our POHLS reduction also has a crucial effect on reducing the computational time to solve these instances.

6 Experiments

In this section, we would like to present more experimental details. All experiments were conducted on a server with Intel Xeon E5-2680 CPU (2.40GHz), 32GB of RAM and Ubuntu 20.04 operating system.

[1] L. Zhu, private communication with F. Ma, July 2020.

Table 1. The spectrum of existence results for the HLS benchmarks. 'Y' indicates that a valid HLS was found, 'N' indicates that such an HLS has been proven not to exist, and 'TO' indicates that the solver failed to make a decision within the time limit (1 week). The new results first presented in this work are marked with *, while the results that can only be solved via symmetry breaking are marked with **.

Order	Problem	Id. (1)	Id. (2)	Id. (3)	Id. (4)	Id. (5)	Id. (6)	Id. (7)
7	HLS($1^5 2^1$)	Y	N	N*	N*	N	N*	N
	HLS($1^3 2^2$)	N*	N*	N*	N*	N*	N*	N*
	HLS($1^1 2^3$)	N	N*	N*	N*	N*	N*	N*
8	HLS($1^6 2^1$)	N	N	N*	N*	N	N*	N
	HLS($1^4 2^2$)	N*	N*	N*	N*	N*	N*	N*
	HLS($1^2 2^3$)	N*	N*	N*	N*	N*	N*	N*
	HLS(2^4)	N	N*	Y*	N*	Y*	N*	N*
9	HLS($1^7 2^1$)	N	N	N*	N*	N	N*	N
	HLS($1^5 2^2$)	Y*	N*	N*	N*	N*	N*	N*
	HLS($1^3 2^3$)	N*	N*	N*	N*	N*	N*	N*
	HLS($1^1 2^4$)	Y	Y*	N*	N*	N*	N*	N*
10	HLS($1^8 2^1$)	N	Y	N*	N*	N	N*	N
	HLS($1^7 3^1$)	Y	Y	N*	N*	N	N*	N
	HLS(2^5)	Y	Y*	N*	N*	N*	N*	N*
	HLS($1^2 2^4$)	N*	N*	N*	N*	N*	N*	N*
	HLS($1^4 2^3$)	Y*	Y*	N*	N*	N*	N*	N*
11	HLS($1^9 2^1$)	Y	Y	N*	N*	N	N**	N
	HLS($1^8 3^1$)	Y	Y	N*	N*	N	N*	N
	HLS($1^5 2^3$)	Y*	Y*	N*	N*	N*	N*	N*
	HLS($1^1 2^5$)	Y	Y*	N*	N*	N*	N*	N*
	HLS($2^4 3^1$)	Y	Y*	N*	N*	N*	N*	N*
12	HLS($1^8 2^2$)	Y*	Y*	N*	N*	N*	TO	N**
	HLS($1^6 3^2$)	Y*	Y*	N*	N*	N*	N*	N*
	HLS($1^4 2^4$)	Y*	Y*	N*	N*	N*	N**	N*
	HLS($2^3 3^2$)	N*	N*	N*	N*	N*	N*	N*
	HLS(2^6)	Y	Y*	N*	N*	N*	N**	N*
	HLS(3^4)	Y	N*	Y*	Y*	N*	N*	Y*
13	HLS($1^9 4^1$)	Y*	Y*	Y*	Y*	N*	N*	Y*
	HLS($1^1 2^6$)	Y	Y*	N*	N*	N*	TO	N**
	HLS($1^7 3^2$)	Y*	Y*	N*	N*	N*	N*	N*
	HLS($1^1 3^4$)	Y*	Y*	N*	N*	N*	N*	N*
	HLS($2^5 3^1$)	Y	Y*	N*	N*	N*	N*	N*
	HLS($1^4 2^3 3^1$)	Y*	Y*	N*	N*	N*	N**	N**
14	HLS($1^{12} 2^1$)	Y*	Y*	N*	TO	TO	TO	TO
	HLS($1^{11} 3^1$)	Y*	Y*	N*	TO	TO	TO	TO
	HLS($2^1 3^4$)	Y*	Y*	N*	N*	N*	N*	N*
	HLS($2^5 4^1$)	Y	Y*	N*	N*	N*	N**	N*
	HLS(2^7)	Y	Y*	Y*	Y*	N**	TO	Y*
	HLS($1^4 2^3 4^1$)	Y*	Y*	N*	N*	N*	N*	N*
Number of Y/N		28/11	26/13	4/35	3/34	1/36	0/34	3/34

6.1 Comparison of At-Most-One Encodings

At-most-one (AMO) clauses are an essential gadget in encoding the ‘alldifferent’ constraint in Latin squares. Therefore, it is necessary to find suitable AMO encodings for solving HLS problems through experimentation. We tested three representative AMO encodings, which are:

- Pairwise encoding: The most straightforward method, which directly restricts that no two literals can both be True. This encoding does not introduce new variables, but it generates $O(n^2)$ clauses.
- Binary encoding [11]: It introduces $O(log\, n)$ Boolean variables and generates $O(n\, log\, n)$ clauses. This encoding strikes a balance between the number of new variables and clauses required.
- Ladder encoding [13]: It builds an efficient inference chain for the AMO constraint by introducing $n - 1$ new variables, which allows the requirement to be satisfied with only $O(n)$ clauses.

The formal description of these encodings can be found in the Appendix.

Table 2. The number of solved instances and the average time taken to solve by Kissat (in seconds) using the three AMO encodings.

Order	#	Pairwise		Binary		Ladder	
		#Solve	$\overline{Time}$	*#Solve*	$\overline{Time}$	*#Solve*	$\overline{Time}$
7	21	**21**	**0.01**	**21**	**0.01**	21	0.02
8	28	**28**	**0.09**	28	0.14	28	0.11
9	28	28	116.08	28	86.34	**28**	**71.58**
10	35	34	274.00	34	165.79	**34**	**126.98**
11	35	34	4952.14	34	3080.61	**34**	**2672.33**
12	42	38	11694.24	**38**	**6109.78**	38	6689.53
13	42	36	18283.23	**38**	**21313.15**	38	21320.52
14	42	30	21617.63	30	16254.64	**31**	**13926.42**
Total	273	249	7759.23	251	6543.86	**252**	**6322.56**

We generate SAT formulae using each of the three AMO encodings, and Table 2 demonstrates the comparison of them. As the order increases, the disadvantage of pairwise encoding becomes apparent compared to the other two, while binary and ladder encodings do not show significant difference in solving time. This indicates that efficient AMO encodings such as ladder and binary can enhance the solving efficiency and potentially find more existence results.

6.2 Comparison with Other Solvers

Apart from propositional logic, the problem can also be modeled as first-order logic formulae over finite domains, and solved by other solvers. Thus, we tried to solve all instances using 3 automated reasoning tools from different backgrounds:

- *MiniZinc* [24]: A powerful constraint programming (CP) platform with the backend solver *Gecode*, which has implemented efficient propagation techniques for the 'alldifferent' constraint.
- *Mace4* [22]: A classical finite model generation tool, which has inherent symmetry breaking ability. However, it has been no longer maintained since 2009.
- *Z3* [23]: A popular SMT (satisfiability modulo theories) solver. The problems are encoded with the EUF (equality and uninterpreted function) theory.

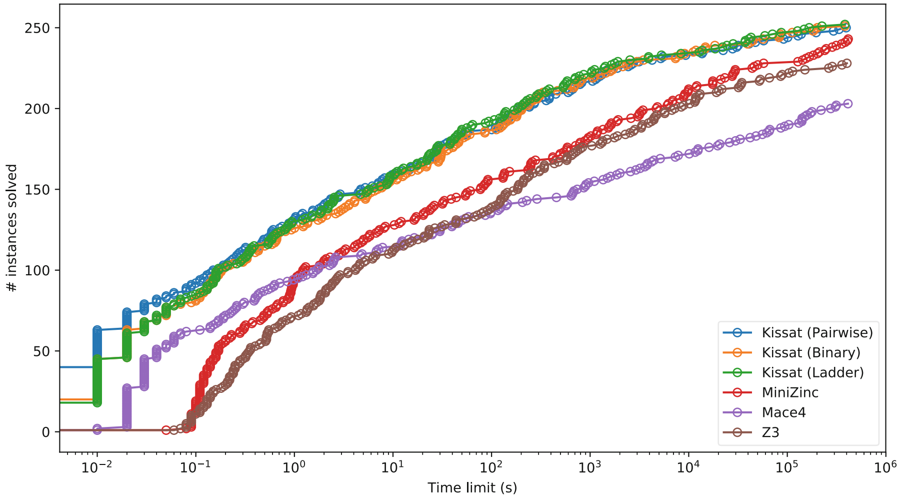


Fig. 3. The number of instances each solver solved within different time limits. Kissat has advantages over other automated reasoning techniques in terms of efficiency.

We record the time consumption of each solver in tackling these instances, as shown in Fig. 3. Each data point represents the number of instances solved by the solver within a particular time limit. The experimental results show that Kissat outperforms other solvers, regardless of which AMO encoding is used. Note that no symmetry breaking clause is added to the SAT formulae at this time. The results indicate that the proposed SAT-based solution is more powerful, which is promising to find more HLSs in a shorter amount of time. Moreover, while Mace4 has some advantages on simple instances (which require <10 s to solve), MiniZinc and Z3 have relatively better performance on harder instances instead.

6.3 Effect of Symmetry Breaking

Although we have implemented efficient encodings and state-of-the-art solvers, some hard benchmarks still remain unsolved. To further improve the efficiency of solving, we have considered two kinds of symmetry breaking approaches in this work. The first one is automatic symmetry breaking tools, which can identify and eliminate symmetries from SAT formula. The second one, as introduced in

Sect. 4, is the POHLS reduction which breaks symmetries from semantics. We employ two automatic symmetry breaking tools, *Shatter* [2] and *BreakID* [9], which can be viewed as preprocessors for the input formulae.

Table 3. The number of instances solved by Kissat using different symmetry breaking techniques within one week. 'Bin' and 'Lad' refer to binary and ladder encoding; 'S' and 'B' refer to Shatter and BreakID, and 'P' denotes the proposed POHLS reduction.

Method	SAT	UNSAT	Total	Method	SAT	UNSAT	Total
Bin	**65**	186	251	Lad	**65**	187	252
Bin+S	**65**	190	255	Lad+S	**65**	188	253
Bin+B	**65**	188	253	Lad+B	**65**	187	252
Bin+P	**65**	190	255	Lad+P	**65**	190	255
Bin+P+S	**65**	**195**	**260**	Lad+P+S	**65**	190	255
Bin+P+B	**65**	193	258	Lad+P+B	**65**	191	256

From Table 3, symmetry breaking has a positive impact on solving unsatisfiable instances, resulting in several new nonexistence results. Figure 4 shows the details of time consumption for solving unsatisfiable instances. It can be observed that the proposed POHLS reduction shows the best performance compared to Shatter and BreakID. Furthermore, we would like to highlight that our POHLS reduction can be used in conjunction with the automatic symmetry breaking tools. As shown in Table 3, the combination of binary encoding, POHLS and Shatter solved the most instances, including two cases: $\mathrm{HLS}^{(6)}(1^9 2^1)$ and $\mathrm{HLS}^{(7)}(1^4 2^3 3^1)$, which are not solvable by all other methods. The results show that our semantic-based POHLS can complement the syntax-based automatic symmetry breaking tools, and successfully improves the solving efficiency.

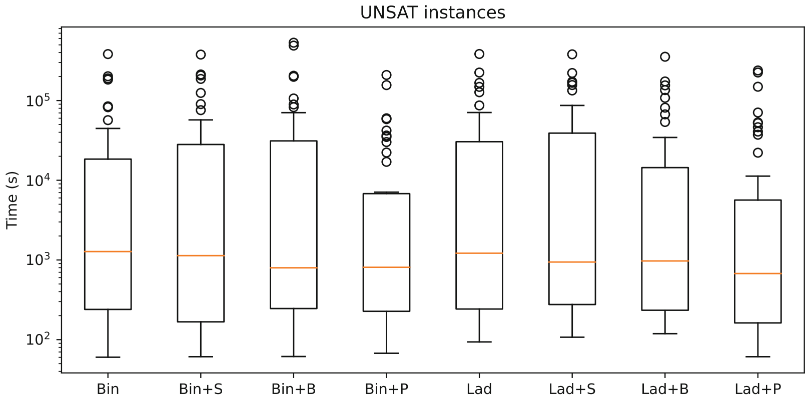


Fig. 4. Time consumption for solving unsatisfiable instances using different symmetry breaking techniques. The instances costing less than 60 s to solve are excluded for clarity.

7 Concluding Remarks

We investigate the existence of holey Latin squares (HLSs) satisfying the seven kinds of identities, which are of special interest to mathematicians. After modeling the existence problem of HLS as propositional logic formulae, a static symmetry breaking method is proposed to further reduce the search space. We utilize efficient SAT solver to test the existence of HLSs in a set of challenging benchmarks, which includes many open cases. Through multiple technical improvements, the existence results of 222 instances are newly reported.

We expect that the methodology and results can serve as the basis for constructing more interesting combinatorial designs, and help to enlighten general existence theorems about HLSs. Another promising direction of future work is to search for more disjoint HLSs based on the single ones we have found, and then discuss the possibility of the existence of corresponding large sets.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (NSFC) under grants No. 62132020 and No. 61972384. Feifei Ma is also supported by the Youth Innovation Promotion Association CAS under grant No. Y202034. The authors would like to thank Lie Zhu at SooChow University for suggesting the open problems and providing help.

References

1. Abel, R.J.R., Li, Y.: Some constructions for T pairwise orthogonal diagonal Latin squares based on difference matrices. Discrete Math. **338**, 593–607 (2015)
2. Aloul, F.A., Markov, I.L., Sakallah, K.A.: Shatter: efficient symmetry-breaking for Boolean satisfiability. In: DAC, pp. 836–839 (2003)
3. Bennett, F.E.: The spectra of a variety of quasigroups and related combinatorial designs. Discrete Math. **77**, 29–50 (1989)
4. Biere, A., Fazekas, K., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT competition 2020. In: Proceedings of SAT Competition 2020 - Solver and Benchmark Descriptions (2020)
5. Bright, C., Cheung, K.K., Stevens, B., Kotsireas, I., Ganesh, V.: A SAT-based resolution of Lam's problem. In: AAAI (2021)
6. Colbourn, C.J.: The complexity of completing partial Latin squares. Discrete Appl. Math. **8**, 25–30 (1984)
7. Colbourn, C.J.: CRC Handbook of Combinatorial Designs. CRC Press, Boca Raton (2010)
8. Colbourn, C.J., Klove, T., Ling, A.C.H.: Permutation arrays for powerline communication and mutually orthogonal Latin squares. IEEE Trans. Inf. Theory **50**, 1289–1291 (2004)
9. Devriendt, J., Bogaerts, B., Bruynooghe, M., Denecker, M.: Improved static symmetry breaking for SAT. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 104–122. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_8
10. Evans, T.: Algebraic structures associated with Latin squares and orthogonal arrays. In: Proceedings of Conference on Algebraic Aspects of Combinatorics (1975)
11. Frisch, A.M., Peugniez, T.J., Doggett, A.J., Nightingale, P.: Solving non-Boolean satisfiability problems with stochastic local search: a comparison of encodings. J. Autom. Reason. (2005)

12. Fujita, M., Slaney, J.K., Bennett, F.: Automatic generation of some results in finite algebra. In: IJCAI, pp. 52–59 (1993)
13. Gent, I.P., Nightingale, P.: A new encoding of alldifferent into SAT. In: International Workshop on Modelling and Reformulating Constraint Satisfaction (2004) (2004)
14. Grant, D.A.: The Latin square principle in the design and analysis of psychological experiments. Psychol. Bull. **45**, 427 (1948)
15. Heule, M.: Schur number five. In: AAAI (2018)
16. Huang, P., Li, R., Liu, M., Ma, F., Zhang, J.: Efficient SAT-based minimal model generation methods for modal logic S5. In: Li, C.-M., Manyà, F. (eds.) SAT 2021. LNCS, vol. 12831, pp. 225–241. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80223-3_16
17. Huang, P., Liu, M., Ge, C., Ma, F., Zhang, J.: Investigating the existence of orthogonal golf designs via satisfiability testing. In: ISSAC (2019)
18. Huang, P., Liu, M., Wang, P., Zhang, W., Ma, F., Zhang, J.: Solving the satisfiability problem of modal logic S5 guided by graph coloring. In: IJCAI (2019)
19. Huang, P., Ma, F., Ge, C., Zhang, J., Zhang, H.: Investigating the existence of large sets of idempotent quasigroups via satisfiability testing. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) IJCAR 2018. LNCS (LNAI), vol. 10900, pp. 354–369. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94205-6_24
20. Lindner, C.C., Stinson, D.R.: Steiner pentagon systems. Discrete Math. **52**, 67–74 (1984)
21. Ma, F., Zhang, J.: Finding orthogonal Latin squares using finite model searching tools. Sci. China Inf. Sci. **56**, 1–9 (2013)
22. McCune, W.: Mace4 reference manual and guide. arXiv preprint cs/0310055 (2003)
23. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78800-3_24
24. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: towards a standard CP modelling language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74970-7_38
25. Pal, S.K., Kapoor, S., Arora, A., Chaudhary, R., Khurana, J.: Design of strong cryptographic schemes based on Latin squares. J. Discrete Math. Sci. Cryptogr. **13**, 233–256 (2010)
26. Parker, E.: Computer investigation of orthogonal Latin squares of order ten. In: Proceedings of the Symposia in Applied Mathematics (1963)
27. Slaney, J., Fujita, M., Stickel, M.: Automated reasoning and exhaustive search: quasigroup existence problems. Comput. Math. Appl. **29**, 115–132 (1995)
28. Zhang, H.: SATO: an efficient prepositional prover. In: McCune, W. (ed.) CADE 1997. LNCS, vol. 1249, pp. 272–275. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63104-6_28
29. Zhang, J., Huang, Z.: Reducing symmetries to generate easier SAT instances. Electron. Notes Theor. Comput. Sci. **125**, 149–164 (2005)
30. Zhang, J., Zhang, H.: SEM: a system for enumerating models. In: IJCAI (1995)
31. Zhang, W., Huang, Z., Zhang, J.: Parallel execution of stochastic search procedures on reduced SAT instances. In: Ishizuka, M., Sattar, A. (eds.) PRICAI 2002. LNCS (LNAI), vol. 2417, pp. 108–117. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45683-X_14

Leiden Fitness-Based Genetic Algorithm with Niching for Community Detection in Large Social Networks

Anjali de Silva(✉), Gang Chen(✉), Hui Ma(✉), and Seyed Mohammad Nekooei(✉)

Victoria University of Wellington, Wellington, New Zealand
{desilanja,aaron.chen,Hui.Ma,mohammad.nekooei}@ecs.vuw.ac.nz

Abstract. Community Detection (CD) in large social networks is a highly active research area due to its immense practical value in many real-world applications. Genetic algorithms (GAs) are widely used to solve the CD problem due to their strong ability to explore the global discrete search space. However, existing GA-based algorithms focus more on the effectiveness of the solution rather than the capability of handling large social networks scalably. In this paper, we propose Leiden Fitness-based GA (LeFGA) to tackle the scalability issue, allowing GA to effectively and efficiently process large social networks. This is achieved specifically by using the newly developed individual and the fitness evaluation method. LeFGA further adopts a niching method to maintain its population diversity. Experimental results prove that LeFGA can significantly outperform multiple state-of-the-art algorithms, especially on large real-world social networks.

Keywords: Community Detection · Leiden · Niching · Large Social Networks

1 Introduction

For many real-world social networks, *communities* are the fundamental building blocks that reveal the underlying architecture of these networks [4]. A *community* is formed by the nodes of the same network N that are tightly connected among each other while loosely connected to the rest of the nodes in N [2,16,21]. The aim of *community detection* (CD) is to identify a *community structure* (CS) that consists of a set of non-overlapping communities. The CD is an important problem for many real-world applications such as product recommendation and criminology [11]. Since a large number of nodes (i.e., more than 100k) can participate in a social network, communities are important for people to gain a comprehensive insight into the functionality of these networks [3]. However, identifying communities within such large networks is challenging because it contains a large number of nodes with complex relationships among each other [2,17].

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 423–435, 2024.
https://doi.org/10.1007/978-981-99-7022-3_39

During the past decade, many interesting approaches have been introduced to effectively detect high-quality community structures of various networks, including heuristic-based approaches [22], mathematical-based approaches [9,10], evolutionary computation (EC)-based approaches [7,8,13,19] and deep learning (DL)-based approaches [21]. Among them, EC-based approaches can achieve a good balance between effectiveness and efficiency [16]. Among all EC techniques, Genetic Algorithm (GA) is widely explored for CD due to its flexibility in solution representation and the capability of maintaining the balance between exploitation and exploration [4].

Leiden-based Genetic Algorithm (LGA) [6] is recently proposed to detect communities in moderate-sized networks. The algorithm adopts the popular Leiden [22] algorithm to improve the chromosomes evolved by GA, which allows LGA to significantly outperform several cutting-edge GA approaches. While LGA achieved impressive results on several benchmark networks, the design of LGA faces some key challenges regarding its scalability and the issue of premature convergence, as explained below.

LGA requires performing an *encoding* operation to map the community structure obtained by the Leiden algorithm (refer to Sect. 4) to an equivalent chromosome (i.e., a chromosome that can be decoded to reproduce the same community structure), which will be further evolved by GA. Referring to Sect. 4, the time complexity of the encoding process as well as LGA is high. As a result, LGA cannot scale well to detect communities in large networks, including popular networks such as Amazon [12] and DBLP [12]. Moreover, LGA converges very fast and can quickly lose its population diversity [6]. This is because the improved solutions obtained by Leiden are very similar and can quickly dominate the rest of other evolved community structures in the GA population, hurting the diversity of the population and resulting in premature convergence.

To address these issues, in this paper, we introduce a new algorithm named Leiden Fitness-based Genetic Algorithm (LeFGA) to make LGA a scalable algorithm while maintaining the population diversity to enhance its effectiveness. LeFGA eliminates the encoding operation of LGA with its newly designed individual and fitness evaluation. According to the computation complexity analysis in Sect. 5, LeFGA is significantly more scalable than LGA. Our experimental results further indicate that LeFGA can outperform several state-of-the-art CD algorithms on multiple benchmark networks. The key contributions of this paper are listed below:

- We propose a novel construction of the individual to be evolved by LeFGA, s.t. $I = (c, CS_L)$, where c is the chromosome evolved by LeFGA and CS_L is the community structure obtained by Leiden based on the initial community structure obtained from the chromosome c. Moreover, the fitness value of the individual is calculated on CS_L. The proposed individual together with the fitness evaluation method reduces the algorithm's complexity as it eliminates the encoding operation of LGA and makes it more scalable.
- Develop a niching method consisting of two components (i.e. *niche creation* and *fitness sharing*) to maintain population diversity throughout the

evolution process. The concept of clustering is used for niche creation, and fitness sharing is used to penalize the fitness values among individuals of the same niche while giving them a fair chance to survive and contribute to the population's diversity.

- Comprehensive experiments have been conducted on widely used real-world benchmark networks. The obtained results clearly show that our proposed algorithm can perform significantly better compared to several state-of-the-art CD algorithms.

2 Related Works

Recently, numerous research has been carried out to solve the CD problem due to its practical importance for many real-world applications [11,21]. This paper focuses on non-overlapping CD in unweighted and undirected large social networks. Among the existing approaches [16,21] for CD, heuristic-based [5], greedy search [22] and EC-based [7,19] approaches have been commonly used to design effective and scalable CD algorithms.

LCDR [1] is one of the heuristic-based algorithms that use the local information of the nodes to identify the core nodes which helps to extract the communities around them. LCDR can process large networks efficiently and is often used as a competing algorithm. Besides LCDR, the Leiden [22] algorithm is another commonly used greedy search algorithm that can scalably identify high-quality community structures. It continuously merges smaller communities into larger communities to increase the modularity (given in Eq. (2)). The merging process is conducted locally and may be potentially trapped by poor local optima. Actually, Leiden is capable of refining a given community structure. If a community structure can be given as the initial community structure for Leiden to perform its merging process, then it is possible for Leiden to find significantly better community structures.

Besides the aforementioned approaches, different EC-based approaches have been proposed for CD [8,16]. A comprehensive review can be found in [16]. As shown in [16], GA is the most commonly studied technique. Several state-of-the-art GA approaches, including LSSGA [7], CCGA [19], and LGA [6], have been developed successfully in the past few years. However, most of the existing GA approaches focus on identifying optimal community structures of small- and medium-scale networks. They face difficulties in processing large social networks efficiently and effectively due to the complexity of the relationship among individuals participating in such networks.

3 Problem Formulation

A social network can be modeled as a graph $N = (V, E)$, where V is the set of n nodes, i.e., $V = \{v_1, v_2, ..., v_n\}$ and E is the set of edges, i.e., $E = \{e_{i,j} | e_{i,j} \in V \times V\}$. This paper considers CD on undirected and unweighted networks.

A *community structure* of a given social network N refers to a set of communities, denoted as $CS = \{C_1, C_2, ..., C_p\}$, $p \geq 1$. In this paper, each community is non-overlapping with any other communities in CS. i.e., $\forall q \neq l$, $C_q \cap C_l = \emptyset$ and $\cup_{q=1}^{p} C_q = V$. The ultimate goal of the CD problem is to identify the community structure CS^* with the maximum modularity defined in Eq. (2). CS^* is defined in Eq. (1).

$$CS^* = \arg\max_{CS} Q(CS) \tag{1}$$

In Eq. (1), $Q(CS)$ refers to the modularity of community structure CS [14] to measure the quality of a given community structure CS. It is the most widely used evaluation metric to determine the effectiveness of a CS, as defined in Eq. (2).

$$Q(CS) = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C_i, C_j, CS) \tag{2}$$

In Eq. (2), m refers to the total number of edges in N, while k_i and k_j refer to the degree of nodes i and j, respectively. A_{ij} is 1 if nodes i and j are adjacent (i.e., connected directly by an edge) or 0 otherwise. $\delta(C_i, C_j, CS)$ returns 1 if nodes i and j belong to the same community or 0 otherwise.

4 Scalability of Leiden-Based Genetic Algorithm (LGA)

This section briefly describes the baseline LGA algorithm [6] to understand its scalability problem which is the motivation behind the new algorithmic development in this paper. LGA proposed to use Leiden to improve the effectiveness of the mutation operator in GA. As one of the most popularly used CD algorithms, Leiden can optimize any CS given as its input. Leiden starts to perform its community refining process (i.e., merging process) based on every community of the given CS.

For Leiden-based mutation, a CS decoded from an evolved chromosome is given as the input for Leiden to optimize. Then CS' improved by Leiden is encoded back to become a mutated chromosome. This encoding operation must produce a mutated chromosome that is equivalent to CS'. Hence the mutated chromosome can be decoded to reproduce CS' precisely. For this purpose, during the encoding process, LGA creates a spanning tree for each community in CS' by conducting the breadth-first search (BFS) to determine a unique parent node of every node in the same community. The time complexity of BFS in the worst case is $O(n^2)$. BFS is executed on every community in CS'. Due to this reason, the complexity of the encoding process is also $O(n^2)$. Consequently, the total complexity of LGA can be expressed as $O(N_p N_g n^2)$, where N_p, and N_g are the population size and the maximum number of generations, respectively. In view of its high time complexity, LGA is hard to scale well for large and complex social networks such as DBLP [12] and Amazon [12].

5 Proposed Algorithm

5.1 Overall Algorithm Design

This section proposes a new algorithm named Leiden Fitness-based GA (LeFGA) that improves the scalability of LGA for CD in large social networks. The fitness evaluation of LeFGA is based on the high-quality community structures produced by Leiden upon giving different initial community structures for Leiden to optimize. As elaborated in Subsect. 5.2, individuals in LeFGA are constructed to hold the chromosome evolved by GA together with the corresponding CS obtained by Leiden. Since the individual explicitly maintains the CS obtained by Leiden, it is unnecessary to encode CS back to the chromosome as in LGA. Since the encoding process is eliminated in the proposed LeFGA, the computation complexity is reduced significantly from $O(N_pN_gn^2)$ to $O(N_pN_gnlogn)$, allowing LeFGA to process large networks scalably. Moreover, to address the premature convergence issue, an effective niching method is proposed to maintain the diversity of the population during the evolution process.

In LeFGA, all the evolved chromosomes follow the Locus-based Adjacency (LBA) representation [15] explained in Subsect. 5.2. The population initialization and the fitness evaluation of each chromosome are performed as described in Subsects. 5.3 and 5.4 respectively. In each generation, we adopt the elitism mechanism that allows a certain number of individuals with the highest fitness to survive directly to the next generation. The rest of the individuals in each generation are created by performing crossover and mutation on the parent individuals, which will be selected randomly based on the shared fitness values using the niching technique developed in Subsect. 5.5. The evolution process is performed iteratively over multiple generations until the termination criteria are reached. Finally, the algorithm returns the best CS with the highest fitness value.

5.2 Construction of the Individual

Traditionally GA directly uses the chromosomes as its individuals in a population. This requires us to adopt a time-consuming encoding operation described in Sect. 4. To reduce computation complexity, LeFGA completely avoids the *encoding* process by keeping the CS obtained by Leiden inside every individual.

Concretely, in LeFGA, an individual I is defined as a tuple $I = (c, CS_L)$, where c is the chromosome evolved by LeFGA and CS_L refers to the community structure optimized by Leiden using CS_c decoded from c (i.e., $CS_c = decode(c)$) as its input s.t. $CS_L = Leiden(N, CS_c)$. Figure 1 illustrates the construction of an example individual in LeFGA.

As illustrated in Fig. 1(b), the chromosome c in any individual evolved by LeFGA consists of n genes where n is the number of nodes of the social network N (e.g., Fig. 1(a)) under processing. The index of each gene refers to a specific node. If the allele value of node v_i is v_j in the chromosome, that means an edge exists between nodes v_i and v_j. Therefore, nodes v_i and v_j should be grouped into the same community according to c. Obeying this rule, LBA performs a decoding

Algorithm 1. Leiden Fitness-based GA (LeFGA)

```
Input: Social Network $N$; Population size $N_p$; Crossover rate $P_c$;
       Mutation rate $P_m$; Elitism ratio $P_e$; Generation size $N_g$;
       Population ratio to generate the offspring $P_s$
Output: Community structure $CS^*$
Generate a population with randomly created $N_p$ chromosomes        ▷ Refer Subsection 5.2
for each chromosome $c$ in $N_p$ do
    Obtain $CS_L$ s.t. $CS_L = Leiden(N, CS_c)$ and $CS_c = decode(c)$
    Construct the individual $I$ s.t $I = (c, CS_L)$                  ▷ Refer Subsection 5.2
    Update the initial population with the individual $I$           ▷ Refer Subsection 5.3
    Evaluate the fitness $f(I)$ of individual $I : Q(CS_L)$           ▷ Refer Subsection 5.4
end for
for each generation $g$ in $N_g$ do
    Pass the best $P_e$ individuals of $N_p$ to generation $g + 1$
    Perform niching method to obtain the shared fitness $F(I)$ values for all $I$
                                                                    ▷ Refer Subsection 5.5
    Update the fitness of each individual $I$ in the population with $F(I)$
    Select parent individuals $N_{OS}$ with the probability of $P_s$ based on $F(I)$
    for each individual $I$ in $N_{OS}$ do
        Perform crossover on pair of chromosomes in two $I$s with a probability of $P_c$ to obtain
        $I_c$                                                        ▷ Refer Subsection 5.6
        Perform random mutation on chromosome $c$ in $I_c$ with a probability of $P_m$ to obtain $I_m$
                                                                    ▷ Refer Subsection 5.6
        Evaluate fitness of individual $I_m$: $Q(CS_L)$                ▷ Refer Subsection 5.4
        Update population with $I_m$ for the next generation $g + 1$
    end for
end for
Return the community structure $CS^*$ obtained by Leiden for the $I$ which has the highest fitness
value: $CS^* = CS_L$
```

process to obtain the corresponding CS_c as given in Fig. 1(c). For example, as given in Fig. 1(b), the allele value of node 1 is node 2, which means node 1 and node 2 belong to the same community. The formation of all communities in CS_c can be determined using this idea. Then, CS_L is obtained by optimizing the CS_c by the Leiden algorithm. Further, modularity values of CS_c (i.e., $Q(CS_c) = 0.2985$) and CS_L (i.e., $Q(CS_L) = 0.3036$) show that Leiden can find CS_L with higher modularity that improves CS_c. Demonstrated by the example individual in Fig. 1(d), LeFGA keeps track of c and CS_L throughout the evolution process. Hence the encoding operation becomes unnecessary.

5.3 Population Initialization

LeFGA randomly generates a set of chromosomes to build its initial population. An example of a randomly generated chromosome is given in Fig. 1(b). According to the social network given in Fig. 1(a), the list of neighbors for node 1 is {2,4}. Hence, the allele value for node 1 is selected randomly from this neighbor list. Every chromosome c is paired with its respective CS_L to form a complete individual, which will be evaluated to determine its fitness. These individuals together become the initial population.

5.4 Fitness Evaluation

LeFGA uses a simple fitness function to perform the fitness evaluation of each individual in the population. Specifically, for any individual $I = (c, CS_L)$,

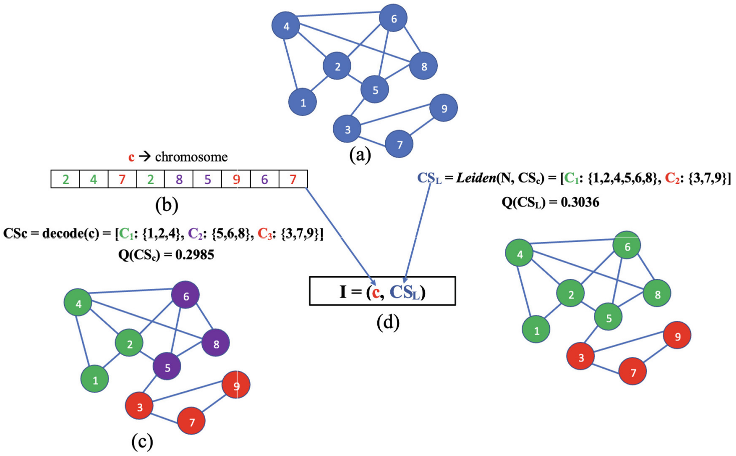

(a)
c → chromosome
2 4 7 2 8 5 9 6 7
(b)
CSc = decode(c) = [C1: {1,2,4}, C2: {5,6,8}, C3: {3,7,9}]
Q(CSc) = 0.2985
(c)
I = (c, CSL)
(d)
CSL = Leiden(N, CSc) = [C1: {1,2,4,5,6,8}, C2: {3,7,9}]
Q(CSL) = 0.3036

Fig. 1. An example individual: (a) an example social network; (b) an example chromosome based on the network given in (a); (c) community structure CS_c obtained through decoding the example chromosome c (different communities in CS_c and CS_L are distinguished by different colors); (d) a complete example of an individual.

LeFGA calculates its fitness as the modularity (Q) (given in Eq. (2)) of CS_L. This fitness function $f(I)$ of individual I is defined in Eq. (3).

$$f(I) = Q(CS_L) \tag{3}$$

According to Eq. (3), the fitness of $I = (c, CS_L)$ can be interpreted as its capability to allow Leiden to generate high-quality CS_L by improving upon the evolved chromosome c. It is worthwhile to note that Leiden's computation complexity is $O(n \cdot \log n)$ [22], implying that it is an efficient and scalable algorithm for CD. Hence, utilizing Leiden in fitness evaluation does not hurt the scalability of the LeFGA algorithm. Once the evolution process is completed, the CS of the fittest individual I will be reported by LeFGA as its best solution for any given social network N.

5.5 Niching Method

Maintaining the population diversity over the generations ensures that the evolved populations can retain good coverage of different regions of the solution space for continued exploration. Since LeFGA uses Leiden-based fitness evaluation, after a few generations, the community structures CS_L found among the evolved individuals may become highly similar, resulting in low population diversity. We propose a niching method to tackle this issue in this subsection.

The niching method is composed of two components. i.e. *niche creation* and *fitness sharing*. The niches are created based on the distance between each individual, where the distance is inversely proportional to the similarity between the individuals of the population. Note that different individuals represent different CSs of the same social network.

We use *Normal Mutual Information (NMI)* to compute the similarity score between any two individuals. The formal definition of NMI is given in Eq. (4).

$$NMI(X,Y) = \frac{-2\sum_{i=1}^{g_X}\sum_{j=1}^{g_Y} P_{ij}\log(P_{ij}n/P_{i.}P_{j.})}{\sum_{i=1}^{g_X} P_{i.}\log(P_{i.}/n) + \sum_{j=1}^{g_Y} P_{.j}\log(P_{.j}/n)} \tag{4}$$

where P represents the confusion matrix, and the element P_{ij} refers to the number of nodes of the community $X_i \in X$ that are also in the community $Y_i \in Y$. The values of i and j span within a range of $\{1,\ldots,n\}$ where n is the number of nodes in N. g_X refers to the number of groups in partition X. g_Y refers to the number of groups in partition Y. $P_{i.}$ denotes the sum of the elements of P in the i-th row and $P_{.j}$ denotes the sum of the elements of P in the j-th column. For any X and Y, NMI(X,Y) falls between 0 (i.e., X and Y are completely different) and 1 (i.e., X is exactly the same as Y).

Based on NMI, we can further create the *distance matrix* $D_{I_iI_j}$ that is formally defined in Eq. (5).

$$D_{I_iI_j} = 1 - NMI(CS_L^{I_i}, CS_L^{I_j}) \tag{5}$$

where $NMI(CS_L^{I_i}, CS_L^{I_j})$ refers to the NMI score (i.e., similarity) between individuals I_i and I_j, $CS_L^{I_i}$ and $CS_L^{I_j}$ denote respectively the community structures found in the two individuals I_i and I_j. If the two individuals are similar (i.e., high NMI score), then their distance should be small and vice versa. Hence we subtract 1 by the NMI value to obtain the distance between any two individuals, as defined in Eq. (5).

After computing $D_{I_iI_j}$, we use a clustering algorithm to identify the niches. For this purpose, we choose the *Density-based spatial clustering of applications with noise* (DBSCAN) algorithm due to several key reasons [20]. DBSCAN has the capability of identifying the clusters without relying on a given number of clusters. Further, DBSCAN is computationally efficient (i.e., $O(n \cdot logn)$) and does not hurt the scalability of LeFGA [20]. LeFGA uses the clusters obtained by DBSCAN among all individuals of a population as its niches. Based on the niches, it further calculates the adjusted/shared fitness value $F(I)$ for each individual I in a niche, according to Eq. (6).

$$F(I) = \frac{f(I)}{s_I} \tag{6}$$

where $f(I)$ refers to the original fitness value of the individual I defined in Eq. (3) and s_I is the niche count that gives the number of individuals in the same niche as individual I. Based on the shared fitness $F(I)$ of all individuals, the rest of the evolution process is done in each generation.

5.6 Genetic Operators

Similar to [7,19], we use the uniform crossover operator to generate offspring chromosomes. The gene values of the offspring are decided according to a randomly generated binary vector. If the binary value of the corresponding gene is 1, then the respective gene value of the first parent is selected, otherwise, the gene value of the second parent is selected. The parents are selected based on the tournament selection with a size of 7 to maintain a good balance between the exploration and the efficiency [19]. Moreover, we use the random neighbor-based mutation strategy, following many existing works [16]. It randomly selects a position in the chromosome that is to be mutated. Then it selects another neighbor randomly from the list of all neighbors of the node at the mutated position.

6 Experiment and Analysis

Subsection 6.1 gives a detailed explanation of the experimented benchmark networks. Four competing algorithms are elaborated in Subsect. 6.2. Parameter settings of LeFGA are reported in Subsect. 6.3. All the experiments were conducted on desktop computers equipped with Intel(R) Core(TM) i7-8700 with 16GB RAM, Python 3.9, and Networkx 3.1.

6.1 Benchmark Networks

Our experiments are performed on multiple widely used real-world social networks, including large networks, e.g. DBLP and Amazon. The details of each benchmark network have been summarized in Table 1.

Table 1. Experimented real-world benchmark networks.

Network	Type	$\|V\|$	$\|E\|$
Karate [18]	Social	34	78
Dolphins [18]	Social	62	159
Polbooks [18]	Social	105	441
Football [18]	Social	115	613
Jazz [18]	Collaboration	198	2742
Ecoli [19]	Biological	418	519
Email [18]	Communication	1005	25571
Cora [18]	Citation	2708	5429
Facebook [19]	Online social	2888	2981
Citeseer [18]	Citation	3312	4732
Protein [19]	Biological	3724	8748
DBLP [12]	Collaboration	317080	1049866
Amazon [12]	Product co-purchasing	334863	925872

6.2 Baseline Algorithms

The performance of the proposed algorithm LeFGA is compared to four state-of-the-art algorithms. LCDR [1] is a heuristic-based algorithm designed for CD in large social networks. Leiden [22] is a greedy search algorithm. CCGA [19] and LGA [6] are recently developed GA approaches for CD.

6.3 Parameter Settings

LeFGA closely follows the parameter setting of CCGA [19] as CCGA was used as the baseline algorithm for the design of LGA. In particular, the population size is 300, the ratio of elitism is 0.05, the probability for crossover is 0.8, the probability for mutation is 0.2, and 200 generations are performed in each run. The algorithm was run 30 times independently. Similarly, all the competing algorithms were executed with the same parameter setting.

Table 2. Average modularity and standard deviation over 30 runs obtained by LeFGA and competing algorithms. Since the CD problem is a maximization problem, the highest mean rank implies the best performance (on several benchmark networks, LeFGA achieved the best results among all competing algorithms; the respective experiment values are bolded). LeFGA does not perform worse than any other competing algorithms across all benchmark networks.

Network	LCDR [1]	CCGA [19]	Leiden [22]	LGA [6]	LeFGA
	$Q_{avg}(std)$	$Q_{avg}(std)$	$Q_{avg}(std)$	$Q_{avg}(std)$	$Q_{avg}(std)$
Karate	0.3707(0.00)	0.4192(2.39e-03)	0.4197(3.01e-04)	0.4198(0.00)	0.4198(0.00)
Dolphins	0.3780(0.00)	0.5158(5.13e-03)	0.5259(2.59e-03)	0.5285(0.00)	0.5285(0.00)
Polbooks	0.5020(0.00)	0.5201(3.57e-03)	0.5269(1.31e-04)	0.5272(0.00)	0.5272(0.00)
Football	0.6020(0.00)	0.5006(2.71e-02)	0.6046(5.48e-05)	0.6046(0.00)	0.6046(0.00)
Jazz	0.4088(0.00)	0.4003(8.26e-03)	0.4444(9.85e-04)	0.4451(0.00)	0.4451(0.00)
Ecoli	0.7634(0.00)	0.7605(5.64e-03)	0.7802(6.49e-04)	0.7815(0.00)	0.7815(0.00)
Email	0.2807(0.00)	0.2657(1.52e-02)	0.4344(5.29e-04)	0.4347(0.00)	**0.4348(0.00)**
Cora	0.7645(0.00)	0.7578(5.48e-03)	0.8220(6.86e-04)	0.8249(1.27e-04)	**0.8250(1.42e-04)**
Facebook	0.8087(0.00)	0.8087(3.70e-05)	0.8087(0.00)	0.8087(0.00)	0.8087(0.00)
Citeseer	0.8125(0.00)	0.8084(5.50e-03)	0.8948(2.78e-04)	0.8969(3.88e-05)	**0.8970(1.63e-05)**
Protein	0.7314(0.00)	0.7313(3.15e-04)	0.7861(4.70e-04)	0.7892(8.56e-05)	**0.7893(6.22e-05)**
DBLP	0.6930(0.00)	0.6533(2.31e-02)*	0.8315(1.04e-03)	0.8349(7.55e-04)*	**0.8359(1.29e-04)**
Amazon	0.7780(0.00)	0.7534(1.43e-02)*	0.9320(1.56e-04)	0.9324(2.16e-04)*	**0.9328(5.40e-05)**
Mean Rank	1.69	1.31	3.11	4.15	**4.73**

* indicates that the respective algorithms couldn't complete their full run of 200 generations within 10 days. Hence the reported results are the best results found after 10 days.

6.4 Experiment Result Analysis

Table 2 compares the performance of LeFGA and all competing algorithms in terms of modularity. The performance of each algorithm is reported as the average obtained modularity (Q_{avg}) along with the standard deviation (std) over 30 independent runs on each benchmark network.

According to Table 2, LeFGA achieved the same average modularity as LGA on small social networks (i.e., less than 1k nodes). This indicates that LeFGA and

LGA can consistently produce high-quality community structures on small networks. Furthermore, on networks with more than 1k nodes (except the Facebook network), LeFGA performed significantly better than all competing algorithms. This implies that LeFGA is able to identify high-quality community structures most of the time on medium and large-scale networks. On the other hand, the Facebook network appears to be simple despite of its large size since all algorithms can manage to obtain the same modularity on this network.

The most interesting observation of the experiments is that LeFGA can handle large networks scalably (i.e., DBLP and Amazon). The algorithm requires running on our Linux desktop for approximately 10 days in order to find high-quality community structures of these two networks. In comparison, we run LGA and CCGA for 10 days without getting good results. In Table 2, CCGA and LGA reported the Q_{avg} and *std* for DBLP and Amazon, based on the results obtained for up to 10 days. This observation confirms that LeFGA can effectively handle large networks and is much more scalable than other GA-based approaches, including LGA and CCGA. Even though LCDR and Leiden are very efficient on large networks, their effectiveness is not as good as LeFGA. The results for the DBLP and Amazon networks reported in Table 2 clearly show that LeFGA achieved higher modularity than Leiden [22] and LCDR [1].

We further conducted the Friedman test with a confidence level of 95% to compare the performance of LeFGA with the other competing algorithms. The obtained mean rank for each algorithm is reported in Table 2. The highest mean rank (i.e., 4.73) was obtained by LeFGA, proving that LeFGA achieved the best performance among all the competing algorithms. Based on these experiment results, it is safe to conclude that LeFGA significantly outperforms several state-of-the-art algorithms on most of the large networks, including DBLP and Amazon networks.

7 Conclusions

In this paper, we developed a novel GA-based algorithm, named LeFGA, to scalably detect high-quality community structures of large social networks. We introduced a new design of the individual in the population that simplifies the fitness evaluation to completely avoid expensive encoding steps. Additionally, a niching approach was proposed to maintain population diversity, further enhancing the reliability and effectiveness of our new algorithm. Comprehensive experiments have been conducted on a wide range of real-world benchmark networks. Our experiment results proved that LeFGA can significantly outperform multiple state-of-the-art algorithms, especially on large social networks. In the future, it is interesting to develop new techniques based on LeFGA to effectively detect community structures in social networks that are changing dynamically across time.

References

1. Aghaalizadeh, S., Afshord, S.T., Bouyer, A., Anari, B.: A three-stage algorithm for local community detection based on the high node importance ranking in social networks. Phys. A **563**, 1–16 (2021)
2. Al-Andoli, M.N., Tan, S.C., Cheah, W.P., Tan, S.Y.: A review on community detection in large complex networks from conventional to deep learning methods: a call for the use of parallel meta-heuristic algorithms. IEEE Access **9**, 96501–96527 (2021)
3. Azaouzi, M., Rhouma, D., Ben Romdhane, L.: Community detection in large-scale social networks: state-of-the-art and future directions. Soc. Netw. Anal. Min. **9**(1), 1–32 (2019). https://doi.org/10.1007/s13278-019-0566-x
4. Behera, R.K., Naik, D., Rath, S.K., Dharavath, R.: Genetic algorithm-based community detection in large-scale social networks. Neural Comput. Appl. **32**, 9649–9665 (2020)
5. Chen, X., Li, J.: Community detection in complex networks using edge-deleting with restrictions. Phys. A **519**, 181–194 (2019)
6. de Silva, A., Chen, A., Ma, H., Nekooei, M.: Genetic algorithm with a novel Leiden-based mutation operator for community detection. In: Aziz, H., Corrêa, D., French, T. (eds.) AI 2022. LNCS, vol. 13728, pp. 252–265. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22695-3_18
7. Guo, X., Su, J., Zhou, H., Liu, C., Cao, J., Li, L.: Community detection based on genetic algorithm using local structural similarity. IEEE Access **7**, 134583–134600 (2019)
8. Hosseini, R., Rezvanian, A.: AntLP: ant-based label propagation algorithm for community detection in social networks. CAAI Trans. Intell. Technol. **5**(1), 34–41 (2020)
9. Jin, D., et al.: ModMRF: a modularity-based Markov random field method for community detection. Neurocomputing **405**, 218–228 (2020)
10. Jing, B.-Y., Li, T., Ying, N., Yu, X.: Community detection in sparse networks using the symmetrized Laplacian inverse matrix (SLIM). Stat. Sin. **32**(1), 1–22 (2022)
11. Karataş, A., Şahin, S.: Application areas of community detection: a review. In: 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, pp. 65–70. IEEE (2018)
12. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection (2014). http://snap.stanford.edu/data
13. Li, X., Wu, X., Xu, S., Qing, S., Chang, P.-C.: A novel complex network community detection approach using discrete particle swarm optimization with particle diversity and mutation. Appl. Soft Comput. **81**, 1–21 (2019)
14. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 1–15 (2004)
15. Park, Y., Song, M., et al.: A genetic algorithm for clustering problems. In: Proceedings of the Third Annual Conference on Genetic Programming, pp. 568–575 (1998)
16. Pizzuti, C.: Evolutionary computation for community detection in networks: a review. IEEE Trans. Evol. Comput. **22**(3), 464–483 (2017)
17. Rajita, B.S.A.S., Kumari, D., Panda, S.: A comparative analysis of community detection methods in massive datasets. In: Goel, N., Hasan, S., Kalaichelvi, V. (eds.) MoSICom 2020. LNEE, vol. 659, pp. 174–183. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-4775-1_19

18. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
19. Said, A., Abbasi, R.A., Maqbool, O., Daud, A., Aljohani, N.R.: CC-GA: a clustering coefficient based genetic algorithm for detecting communities in social networks. Appl. Soft Comput. **63**, 59–70 (2018)
20. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans. Database Syst. **42**(3), 1–21 (2017)
21. Su, X., et al.: A comprehensive survey on community detection with deep learning. IEEE Trans. Neural Netw. Learn. Syst. 1–21 (2022)
22. Traag, V.A., Waltman, L., Van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Sci. Rep. **9**(1), 1–12 (2019)

Non-revisiting Stochastic Search for Automatic Graph Learning

Chenyang Bu(✉) and Hezhen Lu

Key Laboratory of Knowledge Engineering with Big Data (The Ministry of Education of China), School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
chenyangbu@hfut.edu.cn, hezhenlu@mail.hfut.edu.cn

Abstract. In recent years, automatic graph learning (AutoGL) has been widely concerned by academia and industry because it can significantly reduce the threshold and labor cost of graph learning. It has shown powerful functions in hyper-parameter optimization, model selection, graph neural architecture search, and feature engineering. With the development of the network structure, the time and computing resources consumed by the AutoGL process are increasing. AutoGL can be viewed as a bilevel optimization problem encompassing inner and outer optimization. The inner optimization focuses on optimizing the model parameters through techniques like stochastic gradient descent, aiming to minimize the loss function and enhance model performance. On the other hand, the outer optimization aims to identify the best configuration settings for hyperparameters and neural network structures. To address the computational cost associated with heuristic algorithms in bilevel optimization for AutoGL, the non-revisiting idea is proposed to keep a record of all previously evaluated individuals to avoid redundant search. Experiments on multiple search algorithms demonstrate that non-revisiting can improve the time performance of AutoGL. The time performance of the search algorithm with non-revisiting is improved by 40% to 80% compared with that without non-revisiting under the condition that the experimental accuracy remains unchanged.

Keywords: Automatic graph learning · population-based search · non-revisit

1 Introduction

Selecting appropriate hyperparameters (such as learning rates and regularization parameters) and neural network structure (such as the number of layers and the number of hidden units) in traditional machine learning typically requires manual experience and a large number of experiments [9]. This process is time-consuming and often relies on the knowledge of domain experts. Automated graph machine learning (AutoGL) aims to discover optimal hyperparameter and neural network structure configurations for different graph tasks or graph data

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 436–447, 2024.
https://doi.org/10.1007/978-981-99-7022-3_40

without manual design [2]. There are various structures of graph neural networks, among which the typical ones are GCN and GAT [10]. AutoGL has received increasing attention due to its ability to significantly reduce the threshold and human cost of graph learning.

From an optimization point of view, AutoGL can be regarded as a bilevel optimization problem consisting of inner optimization and outer optimization [4]: 1) Inner optimization focuses on learning and optimizing the model parameters (e.g., the weights of the neural network) [11], by training the model on the given data using the specified hyperparameters and neural network structure configuration. Typically, traditional optimization methods like stochastic gradient descent are employed to solve these inner optimization problems. The goal is to minimize the loss function and improve the model performance through parameter updates. 2) Outer layer optimization involves finding the best configuration settings in the search space of hyperparameters and neural network structures to achieve the optimal solution for the inner optimization problem. This can be accomplished using various methods such as grid search, random search, Bayesian optimization, and evolutionary algorithms. By treating automatic graph machine learning as a bilevel optimization problem, we can automate the process of discovering the optimal hyperparameter and neural network structure configurations. It should be pointed out that in this two-layer optimization problem, each fitness evaluation in the outer layer optimization problem corresponds to a model training process in the inner layer optimization problem. This implies that the optimization process can be computationally expensive [3].

Bilevel optimization has received extensive attention in the field of machine learning [6]. Existing methods for solving bilevel optimization problems mainly include the following categories: analytical methods, gradient descent methods, and heuristic methods. 1) Analytical methods refer to finding closed-form solutions by solving the analytical expression of the problems. This kind of method requires the problem to have a simple structure and the mathematical model is known. 2) Gradient descent methods update the parameters using the gradient information through the backpropagation algorithm. This type of method typically requires the function to be linear, or strongly convex to guarantee convergence to the global optimum. 3) Heuristic algorithms are suitable for solving complex, high-dimensional or nonlinear optimization problems that are difficult to solve directly using traditional analytical methods or gradient descent methods. Typical heuristic algorithms such as evolutionary algorithms [1], has the advantages of parallelization and adaptability, and have shown good performance in many practical problems.

In response to the computational cost associated with heuristic algorithms for bilevel optimization, some researchers have attempted to combine heuristic approaches with classical optimization methods. This integration aims to leverage the strengths of both approaches and further reduce computational time by employing surrogate models. Surrogate models refers to the utilization of a simplified model or function to approximate the value of the objective function, thereby mitigating the need for frequent evaluations of the actual objective function.

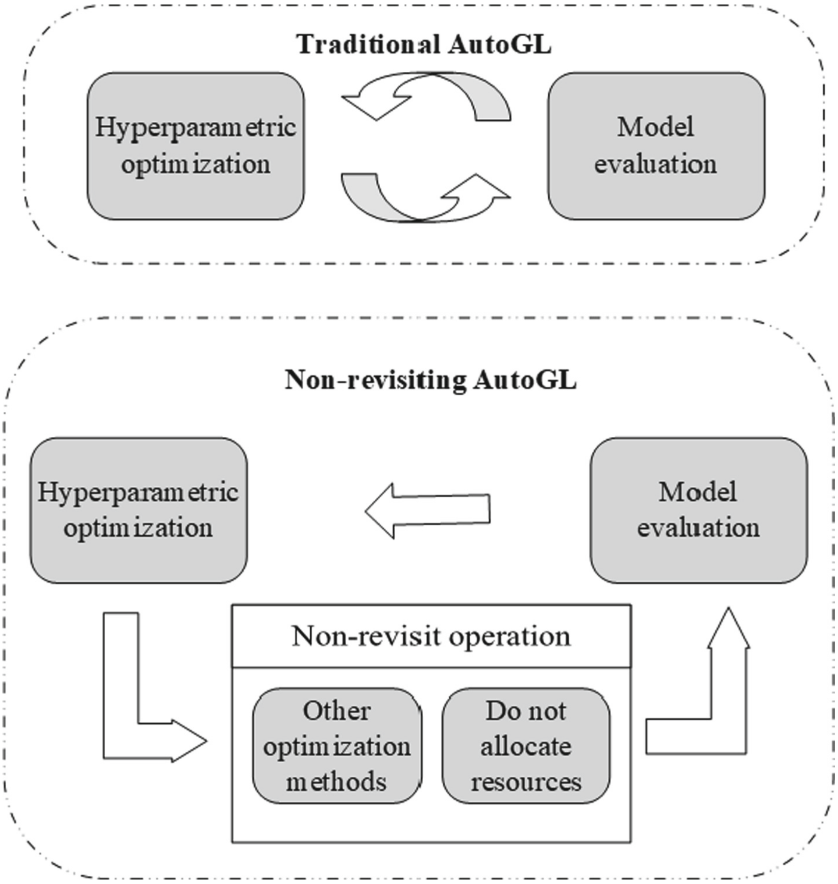

Traditional AutoGL
Hyperparametric optimization
Model evaluation
Non-revisiting AutoGL
Hyperparametric optimization
Model evaluation
Non-revisit operation
Other optimization methods
Do not allocate resources

Fig. 1. Traditional AutoGL process and non-revisiting AutoGL process

By employing the surrogate model, the computational cost associated with evaluating the objective function can be significantly reduced (Fig. 1).[1]

Due to the lengthy training time of the model, the number of sample points may be limited, which can cause the failure of neural network-based methods. An intuitive approach to constructing a surrogate model is to keep a record of all previously evaluated individuals to avoid redundant searches. However, storing this information using arrays or similar methods can be inefficient. Hence, in this study, to reduce the time consumption of heuristic optimization methods for automated graph learning, we propose a non-revisiting mechanism that can be applied to any population-based search method.

In summary, the contributions of this study are as follows:

[1] https://github.com/710965953/Non-revisiting.

1. This is the first time that the non-revisiting idea has been applied to the field of automatic machine learning. By using the real history model and its performance saved in the non-revisiting mechanism, we can reduce the resources wasted due to the repeated evaluation of similar models in AutoGL.
2. We apply the non-revisiting mechanism to five classical evolutionary algorithms and the hyperparameter optimization of two representative graph neural networks. The experimental results on three real datasets demonstrate that, for the tested evolutionary algorithms and tasks, the search algorithm utilizing the non-revisiting mechanism exhibits superior time performance compared to the search algorithm without the non-revisiting mechanism, without statistically significant reductions in accuracy.

The remaining sections of this paper are organized as follows. Section 2 presents the relevant background information. Section 3 introduces the proposed mechanism. Section 4 describes the experimental setup and results. Finally, the paper concludes with a summary of the findings.

2 Background

In this section, we introduce the background of bilevel optimization and automatic graph learning.

2.1 Bilevel Optimization

The bilevel optimization problem originally originated from the field of economic game theory [7], involving a hierarchical form of mathematical programming. The feasible domain of one optimization task is restricted by the solution set mapping of another optimization task, with the latter embedded within the former. The upper-level problem denotes the external optimization task, while the lower-level problem refers to the internal optimization task. Bilevel optimization problems have found numerous applications in intricate machine learning problems, including hyperparameter optimization, meta-learning, neural architecture search, and deep reinforcement learning [4,7]. Consequently, these problems have garnered significant attention within the machine-learning community.

2.2 Automatic Graph Learning

In recent years, deep learning has shown excellent performance in various fields and has been used by researchers to solve many challenging tasks. At the same time, to achieve better performance, the structure of deep neural networks has become more and more complex. For example, VGG-16 [8] has more than 130 million parameters, occupies nearly 500 MB of memory, and requires 15.3 billion floating-point operations to process the input image of size 224 × 224. However, such models are developed by human experts with a lot of trial and error, that is, even experts need a lot of resources and time to design a good model. To

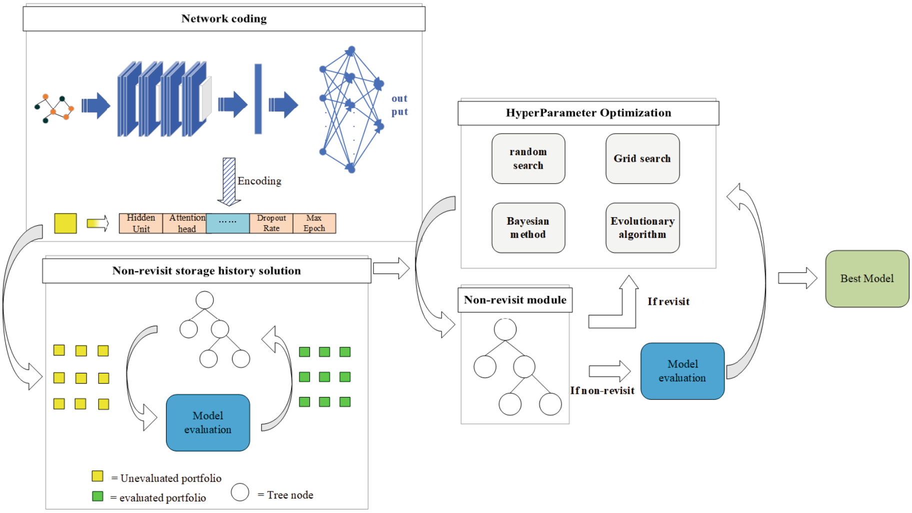


Fig. 2. Model architecture diagram. In the Network Encoding Module, the model's hidden layer units, number of attention heads, and maximum epoch are encoded and stored. In the Non-revisiting Storage History Solution Module, all evaluated models and their evaluation values are stored in the non-revisit module. In the Hyperparameter Optimization Module, the non-revisitation mechanism and optimization algorithm work together until a satisfactory model is obtained.

reduce these heavy development costs, the idea of automated machine learning (AutoML) [3] has emerged, which refers to the automation of the entire machine learning workflow from model building to application. Compared with some general-purpose machine learning workflows, automated machine learning can achieve comparable or better results than human experts with little or no human intervention. Therefore, automated machine learning can lower the threshold for algorithm learning and use, and is of great help to the application of machine learning algorithms in practical scenarios. Traditional AutoML methods are mainly used to process structured data (such as one-dimensional time series signals and two-dimensional images).

3 Algorithm

The basic idea of the non-revisiting mechanism is to use a BSP tree to store the positions and fitness values of all previously evaluated individuals, as shown in Fig. 2, which helps determine if a new individual needs to be re-evaluated. This reduces the redundancy of evaluating similar or identical individuals, thereby saving computational resources. This section introduces the tree-building process, non-revisiting process, and potential advantages compared to other surrogate models.

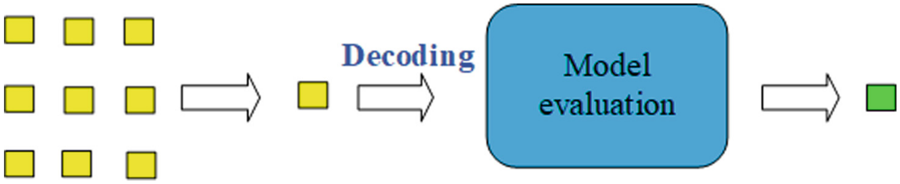


Fig. 3. GNN individual decoding evaluation process. In this process, the model structure and hyperparameters in the chromosome will be reduced to a complete model, and a model evaluation value will be obtained after training.

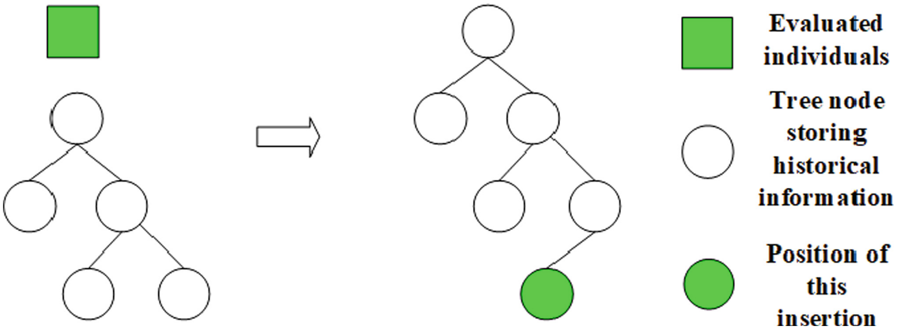

Evaluated individuals
Tree node storing historical information
Position of this insertion

Fig. 4. BSP tree preservation history evaluation solution

3.1 Tree Building Process

We use a BSP tree to partition the search space and store all historically evaluated individuals. Traditional methods for storing memory individuals include array-based methods and tabu lists. However, these existing methods have drawbacks, especially from the perspective of surrogate models.

Binary Space Partitioning (BSP) refers to the method of recursively dividing a search space into two subspaces using hyperplanes. Each node in the tree corresponds to a subregion in the search space. Based on the assumption that the search space has smoothness, we assume that individuals with similar positions have similar fitness values. Therefore, for all unevaluated individuals within a region, we approximate their fitness values with the fitness values of the evaluated individuals in that region. As more individuals are evaluated, the prediction error of the surrogate model may gradually decrease.

The basic structure and construction process of the BSP tree are as follows, and shown in Figs. 3 and 4. All non-leaf nodes in the BSP tree are virtual nodes used for assisting search, while leaf nodes are used to store all evaluated individuals. Each node stores three attributes: the position x of the individual, its fitness value f_x, and the dimension dim along which the subregion is partitioned, represented as $node = k, f_k, dim$. Initially, the tree is empty, and the first inserted node becomes the root. For each new evaluated individual $Z = \{Z_1, Z_2, ..., Z_j, ..., Z_k\}$, we find the evaluated individual X in the BSP tree that is closest to Z based on distance similarity. We then use a hyperplane in the j-th dimension to partition the region where node X is located into two parts, where $j = \arg\max_{j} |Z_j - X_j|$. After the region is partitioned, the original region X is divided into two new subregions, X and Z, which are stored in the left and right child nodes, respectively. The parent node is represented as a virtual node X' (with the same position and fitness as its left child node X).

3.2 Non-revisiting Process

After the BSP tree is constructed, newly generated individuals resulting from crossover and mutation undergo a non-revisiting check before evaluation. The basic idea is to determine the distance between the new unevaluated node and the evaluated nodes in the BSP tree. If the distance is less than a certain threshold, we consider the evaluation of the new individual unnecessary. The specific details are as follows.

For a new unevaluated solution v_i, we use *cru* to denote the current search node. We start the search from the root node of the BSP tree and use the following method to find similar nodes until *cru* reaches a leaf node. If the final solution *cru* found through the search is similar to v_i (i.e., the Euclidean distance is less than the threshold), it means that v_i does not need to be re-evaluated.

$$cru = \begin{cases} Left\ Child\ of\ cru & if\ v_{i,j} < cru_j, \\ Right\ Child\ of\ cru & otherwise \end{cases} \tag{1}$$

where j represents the specific dimension along which the search space was partitioned using hyperplanes during the tree construction phase (the stored j may differ for each node).

4 Experiments

4.1 Experimental Setup

Three famous datasets were used in the experiment, namely Cora, Citeseer, and Pubmed. The statistics of these datasets is given in Table 1. Two representative models of graph neural networks, i.e., GCN and GAT, were selected as the base models. The hyperparameters along with their search ranges are as follows: The Learning Rate ranges from 0.01 to 0.05; Weight Decay Rate ranges from 0.0001 to 0.001; Dropout Rate ranges from 0.2 to 0.8; Number of Hidden Units is a discrete value with a range of {4, 5, ..., 16}; Number of Attention Heads is a discrete value with a range of {6, 8, 10, 12}; Activation Function is a discrete value with a range of {leaky relu, relu, elu, tanh}; Max Epoch is a discrete value with a range of {100, 101, ..., 300}; Early Stopping Round is a discrete value with a range of {10, 11, ..., 30}. The relevant settings for the evolutionary algorithm are as follows: Population size is 100, Max gen is 20, Mutation rate is 0.5, and Crossover rate is 0.7, the specific implementation of EA comes from [5].

Table 1. Statistics for the datasets

Dataset	Category	Nodes	Edge	Characteristic
Cora	7	2708	5429	1433
Citeseer	6	3327	4732	3703
Pubmed	3	19717	44338	500

Table 2. Experimental results on the Cora dataset

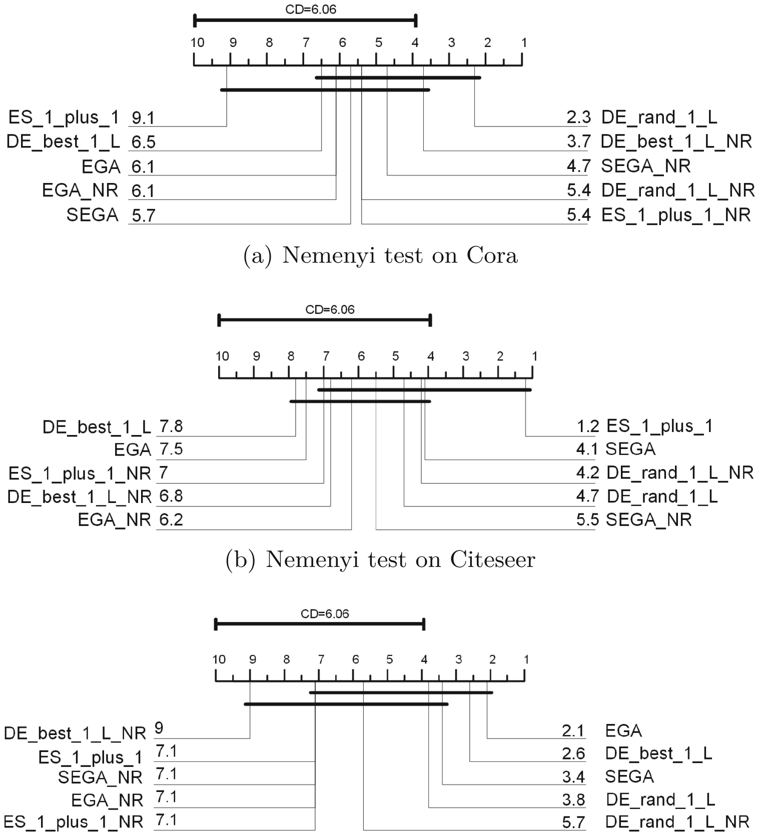
Model	Algorithm	Original version		Non-revisit		Time improvement
		Time	ACC	Time	ACC	
GCN	DE-best-1-L	2501.4 s	0.831	1026.8 s	0.8286	58.9%
	DE-rand-1-L	3226.4 s	0.826	1245.8 s	0.8308	61.3%
	ES-1-plus-1	2713 s	0.836	1237.4 s	0.8308	54.3%
	EGA	2307.2 s	0.8316	440.6 s	0.8314	80.9%
	SEGA	2184.8 s	0.8268	542.8 s	0.829	75.1%
GAT	DE-best-1-L	3096.2 s	0.8414	1531.6 s	0.8376	50.5%
	DE-rand-1-L	2951 s	0.8324	1848.8 s	0.8356	40.0%
	ES-1-plus-1	2875.6 s	0.8356	1984.6 s	0.8352	42.1%
	EGA	2806.8 s	0.8302	799 s	0.8356	71.5%
	SEGA	2733.8 s	0.8368	765.8 s	0.8412	71.9%

4.2 Experimental Results

To verify the time performance and accuracy of AutoGL after incorporating the non-revisit mechanism, this section selects three datasets (Cora, Citeseer, and Pumbed) as target tasks and conducts experiments on two graph neural network models, GCN and GAT. The data sizes of Cora, Citeseer, and Pumbed progressively increase. To validate the effectiveness of the non-revisit mechanism, we conducted experiments on five evolutionary algorithms within AutoGL. The results demonstrate that the approach utilizing the non-revisit mechanism significantly improves the time performance of the model without sacrificing accuracy, as compared to the original evolutionary algorithms without the mechanism. A detailed analysis of the experimental results is provided below.

Comparative results regarding average running time in Tables 2, 3 and 4 indicate that the non-revisit mechanism saves at least 40% of the time in all cases. This aligns with our expectations as the mechanism avoids redundant evaluations of identical or similar models. For instance, in the case of the five evolutionary algorithms on the Citeseer dataset with the GCN model, ES-1-plus-1 reduces the time consumption by 40%, while EGA reduces it by 78%. These data further support our argument that the integration of the non-revisit mechanism with AutoGL improves its time performance.

ACC data in Tables 2, 3 and 4 demonstrate that the accuracy of AutoGL does not decrease when the non-revisit mechanism is added. For example, in the Cora dataset and GAT model, the acc of the ES-1-plus-1 algorithm without the non-revisit mechanism is 0.8356, while it is 0.8352 with the mechanism added. Similarly, in the larger Pumbed dataset and GAT model, the original acc of the ES-1-plus-1 algorithm is 0.7846, which increases to 0.7882 after incorporating the non-revisit mechanism. These data indicate that the accuracy of the original algorithm and the algorithm with the non-revisit mechanism remains within the same range, supporting our argument that the non-revisit mechanism does not compromise the accuracy of AutoGL.



(a) Nemenyi test on Cora

(b) Nemenyi test on Citeseer

(c) Nemenyi test on Pumbed

Fig. 5. Nemenyi test on the accuracy metric for the GCN tasks. A larger ranking indicates a better performance. Models on the same horizontal line have similar predictive performance. The EA with non-revisit mechanism and the EA without non-revisit mechanism are in the same level interval in terms of testing accuracy.

Table 3. Experimental results on the Citeseer dataset

Model	Algorithm	Original version		Non-revisit		Time improvement
		Time	ACC	Time	ACC	
GCN	DE-best-1-L	2646.4 s	0.7278	1259.8 s	0.728	52.3%
	DE-rand-1-L	2652 s	0.7226	1504.6 s	0.7222	43.2%
	ES-1-plus-1	2698.4 s	0.716	1618.2 s	0.727	40.0%
	EGA	3066.2 s	0.7286	671.8 s	0.7264	78.0%
	SEGA	2705.2 s	0.7218	735.8 s	0.7244	72.8%
GAT	DE-best-1-L	3181.2 s	0.7318	1654.4 s	0.7258	47.9%
	DE-rand-1-L	3080.2 s	0.7182	1856.6 s	0.7172	41.0%
	ES-1-plus-1	2962.6 s	0.7228	1755.4 s	0.7208	40.7%
	EGA	4910.8 s	0.729	668.2 s	0.7116	86.3%
	SEGA	2796.6 s	0.7106	741.2 s	0.7136	73.4%

Table 4. Experimental results on the Pubmed dataset

Model	Algorithm	Original version		Non-revisit		Time improvement
		Time	ACC	Time	ACC	
GCN	DE-best-1-L	2288.6 s	0.788	913.4 s	0.7922	60.0%
	E-rand-1-L	2506.2 s	0.789	1344.8 s	0.7892	46.3%
	ES-1-plus-1	2499 s	0.79	1366.6 s	0.79	45.3%
	EGA	2562 s	0.786	556.2 s	0.79	78.2%
	SEGA	2264.8 s	0.7884	755.6 s	0.79	66.6%
GAT	DE-best-1-L	4317.6 s	0.788	2359.6 s	0.7904	45.3%
	DE-rand-1-L	4221 s	0.7818	2446.6 s	0.7908	42.0%
	ES-1-plus-1	4434 s	0.7846	2619.4 s	0.7882	40.9%
	EGA	5022.8 s	0.7892	1311 s	0.7864	73.8%
	SEGA	4985 s	0.7896	1020.4 s	0.7878	79.5%

Among the six sets of data in Tables 2, 3 and 4, the genetic algorithm EGA consistently exhibits the greatest improvement in time performance compared to SEGA. For instance, in the Cora dataset and GCN model, the time improvement of both algorithms is 80.9% and 75.1%, respectively. In the Citeseer dataset and GAT model, the time improvement of both algorithms is 86.3% and 73.4%, respectively. In the Pumbed dataset and GCN model, the time improvement of both algorithms is 78.2% and 66.6%, respectively. One possible reason is that offspring individuals in genetic algorithms inherit more genes from parent individuals, increasing the likelihood of similarity between the generations.

The comparative results of the Nemenyi test on the ACC metric can be found in Fig. 5, where the average ranks of each algorithm are labeled along the axis (lower ranks on the right side). In the Nemenyi test, if the average ranks of two models differ by at least a critical difference (CD), significant differences are considered to exist. The critical difference is calculated using a significance level of 5%. In Fig. 5, models on the same horizontal line exhibit similar predictive performance. For example, in case (b), EGA has a rank of 7.8, while EGA_NR (NR indicating the use of the non-revisitation mechanism) has a rank of 6.8, which does not differ by more than one CD value. Therefore, there is no statistically significant difference between the version with and without the non-revisitation mechanism of the algorithm in terms of the ACC metric. This also validates our conclusion that the non-revisitation mechanism, which improves time performance, does not statistically reduce the accuracy of the original algorithm.

5 Conclusion

In this study, we propose the integration of the non-revisiting mechanism with evolutionary algorithms for application in AutoGL. Extensive experiments were conducted on three real datasets, namely Cora, Citeseer, and Pubmed. The experimental results demonstrate that AutoGL with the non-revisiting mechanism achieves significant improvements in time performance, ranging from 40% to 80%, without statistically significant differences in algorithm accuracy compared to the original algorithm. In future research, we plan to extend the application of the non-revisiting mechanism to additional evolutionary algorithms and propose operators based on the non-revisiting mechanism. These endeavors aim to further enhance the efficiency and effectiveness of automated machine learning techniques.

References

1. Bu, C., Lu, Y., Liu, F.: Automatic graph learning with evolutionary algorithms: an experimental study. In: Pham, D.N., Theeramunkong, T., Governatori, G., Liu, F. (eds.) PRICAI 2021. LNCS (LNAI), vol. 13031, pp. 513–526. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-89188-6_38
2. Guan, C., et al.: AutoGL: a library for automated graph learning. In: ICLR 2021 Workshop on Geometrical and Topological Representation Learning, vol. abs/2104.04987 (2021). https://openreview.net/forum?id=0yHwpLeInDn
3. He, X., Zhao, K., Chu, X.: Automl: a survey of the state-of-the-art. Knowl. Based Syst. **212**, 106622 (2021)
4. Hospedales, T.M., Antoniou, A., Micaelli, P., Storkey, A.J.: Meta-learning in neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5149–5169 (2022)
5. Jazzbin, E.: Geatpy: the genetic and evolutionary algorithm toolbox with high performance in python (2020)
6. Ji, K., Yang, J., Liang, Y.: Bilevel optimization: convergence analysis and enhanced design. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 4882–4892. PMLR (2021)

7. Liu, R., Gao, J., Zhang, J., Meng, D., Lin, Z.: Investigating bi-level optimization for learning and vision from a unified perspective: a survey and beyond. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 10045–10067 (2022)
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. 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). https://arxiv.org/abs/1409.1556
9. Wang, X., Zhu, W.: Automated machine learning on graph. In: Zhu, F., Ooi, B.C., Miao, C. (eds.) KDD 2021: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, 14–18 August 2021, pp. 4082–4083. ACM (2021)
10. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020)
11. Zöller, M., Huber, M.F.: Benchmark and survey of automated machine learning frameworks. J. Artif. Intell. Res. **70**, 409–472 (2021)

Detecting AI Planning Modelling Mistakes – Potential Errors and Benchmark Domains

Kayleigh Sleath and Pascal Bercher(✉)

School of Computing, The Australian National University, Canberra, Australia
{kayleigh.sleath,pascal.bercher}@anu.edu.au

Abstract. AI planning systems can solve complex problems, leaving domain creation as one of the largest obstacles to a large-scale application of this technology. Domain modeling is a tedious, error-prone and manual process. Unfortunately, domain modelling assistance software is sparse and mostly restricted to editors with only surface-level functionality such as syntax highlighting. We address this important gap by proposing a list of potential domain errors which can be detected by problem parsers and modeling tools. We test well-known planning systems and modeling editors on models with those errors and report their results.

Keywords: Automated Planning · Modelling support · Knowledge Engineering · PDDL Modeling · HDDL Modeling

1 Introduction

Automated planning, a branch of artificial intelligence (AI), is concerned with generating sequences of actions that turn one state of a system into a desired one. This requires a formal specification of the planning model, which is written in text files and adheres to a specific syntax such as the planning domain description language (PDDL) [3] for classical (non-hierarchical) planning and its extension HDDL [6] for hierarchical task network (HTN) planning [1].

Although modeling is a complex and error-prone task, the few existing modeling tools focus on syntax highlighting, integrating a planner, and visualizing solutions – but they are of limited use if the domain modeler makes mistakes. Planning systems' parsers provide even less support for detecting such errors; in many cases they just crash, or even worse they don't find solutions or find wrong or inconsistent ones. There are a few more evolved works for modeling support [7], but they all assume a syntactically correct model and are hence orthogonal to our contributions.

To address this problem, we make the following contributions: *(1)* We provide a compilation of potential modeling errors. *(2)* We supply a public repository of 56 (flawed) benchmark domains containing each of these errors, to the best of our knowledge the first benchmark database for AI modeling support. *(3)*

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 448–454, 2024.
https://doi.org/10.1007/978-981-99-7022-3_41

We conduct an evaluation of well-known AI planning tools for their ability to diagnose those errors, showing that not a single tool is able to spot all errors, with no tool being strictly stronger than another.

2 AI Planning Formalism

Due to space restrictions we do not provide a formal introduction to the description languages PDDL [3] and HDDL [6] or their underlying formalisms [1] and only refer to the respective literature. Instead, we explain the input languages based on a PDDL example taken from the PDDL textbook [3].

Listing 1.1. A PDDL action for moving a truck between locations [3].

```
(:action drive
  :parameters (?t - truck ?from ?to - location)
  :precondition (at ?t ?from)
  :effect (and (not (at ?t ?from)) (at ?to)))
```

Classical planning evolves around the transition of states, finite set of facts, propositions that encode what's currently true. States are changed by actions (see Listing 1.1), which have preconditions (here that the truck is at the location **?from**) and effects, specifying how the respective state changes (here that the truck is not at **?from** anymore but at **?to**). Problems are defined in a "lifted" fashion, where variables (parameters) are used to abstract away from concrete constants such as specific trucks or locations. These constants/objects are given in the problem description so that actions can be instantiated as required.

Hierarchical planning adds further constraints [1]. Here, we are additionally given a set of compound tasks and a set of decomposition methods that specify how these tasks could be refined into more primitive tasks and finally into actions. This process is quite similar to formal grammars, where production rules (corresponding to decomposition methods) are used to turn non-terminal symbols (compounds tasks) into terminal symbols (actions). The goal is to turn an initially given task network – a partially ordered sequence of tasks – into an executable action sequence (just as in classical planning), but now tasks can only be obtained by adhering to the hierarchy defined by the decomposition methods.

3 Potential Errors in Planning Domains

This section details a list of errors or potential errors that may be encountered when modelling planning domains in PDDL/HDDL, separated into:

- syntax errors: these are actual errors, but often not spotted by parsers and
- semantics errors: these would be warnings as they indicate a *potential* modeling error, to be checked by the domain modeler.

All errors we identify for classical planning (PDDL) naturally transfer to HTN planning (HDDL) as well, whereas HTN errors are unique to HTN planning. We hence present the respective flaws in two different lists.

The list has been translated into a repository[1] which we see as a first step towards a public testbed for PDDL and HDDL parsers. *We invite others to add additional cases we might not have thought of.*

3.1 Syntax Errors

(1) Inconsistent Parameter Use. The modeller attempts to use a predicate or task with either a parameter of an incompatible type or a different number of parameters than it was defined with. This second error is only possible in HTN planning since in classical planning actions are only defined once (and thus never referenced anymore), whereas HTN planning could re-use a task (primitive or compound) multiple times in decomposition methods.

(2) Undefined Entities. The modeller attempts to use an undefined predicate, type, or task. (Again "using undefined tasks" is only possible in HTN planning for the same reason as mentioned above).

(3) General Syntax Errors. The modeller forgets to include a key piece of syntax or makes a typo - for instance, forgets to write ":parameters" in a task definition (which lists the sequence of typed task parameters), adds an extra parenthesis, forgets a dash when defining a variable, or forgets to write a questionmark in front of a variable name to differentiate it from a constant. It is expected that most of these errors are captured by any parser, but not all are, and useful error messages are not always produced.

(4) Duplicated Definitions. The modeller repeats some definitions (e.g., some task, decomposition method, predicate, or constant). Closely related, the modeller writes duplicate entries in a task definition – for example, includes multiple ":parameters" entries.

(5) Cyclic Type Declaration. When two types are directly or indirectly declared to be subtypes of each other, forming a cycle.

(6) Undeclared Parameters. The modeller tries to use a variable in the definition of a task (or decomposition method in case of HTN planning) that wasn't declared as a parameter of that task (or method).

(7) Cyclic Ordering Constraints. Task networks are defined over a partial ordering – which excludes cycles. – **HTN-specific**

(8) Duplicate Orderings. A method contains both the "ordered subtasks" keyword (which implies that only a *sequence* of tasks is provided), but also a (thus redundant) set of explicit ordering constraints. – **HTN-specific**

[1] https://github.com/ProfDrChaos/flawedPlanningModels.

3.2 Semantic Errors

These are *potential* errors, which do not contradict PDDL/HDDL.

(9) Complementary Effects. There is an intersection between the ground negated and positive effects of a task.

(10) Unsatisfied Preconditions. Some action's preconditions can never be fulfilled. This may be due to syntactically complementary preconditions (with identical predicates, including parameters), or simply since in the given planning problem the precondition can't be made true. While the first possible cause is a simple syntax check the second involves complex reasoning, which is as hard as planning.

(11) Unused Elements. The modeller defines a type or predicate or a parameter in a task (or decomposition method in case of HTN planning) that is not used. In the case of HTN planning, tasks may be "unused", which can be defined as being unreachable from the initial task network.

(12) Redundant Effects. Some effect will never change the state to which the respective action is applied. There are two possibilities how this can happen: The simplest case is if some effect also occurs as a precondition (with identical parameters). The redundancy can however also be problem-dependent, i.e., if any grounding of some effect is contained in any state in which the respective action is applicable.

(13) Immutable Predicate. A predicate is defined which never occurs in task effects. This means the state of that predicate is constant.

(14) Compound Tasks Without a Primitive Refinement. The modeller defines a compound task which can never be refined into a primitive plan (it is therefore useless). A special case of this is not providing any decomposition method for some compound task. – **HTN-specific**

4 Evaluation of Existing Parsers

From the proposed list, we created a large benchmark set with flawed domains. We tested some of the best-known AI planning tools (planning system parsers or domain editors) that parse PDDL and HDDL domains and evaluated their performance. These tools were: editor.planning.domains [9], Visual Studio PDDL Plugin [2], Fast Downward [4], PANDA [5], HyperTensioN [8], and LiloTane [10].

4.1 Results

The software was evaluated for each flawed domain based on three categories:

Error Detection. Whether the software recognises the error and stops the parsing process (‘yes’ – green), crashes without catching the error (‘crashes’ – yellow), or provides a solution/reports unsolvable despite the model being wrong (‘no’ – red).

Location Guidance. Whether the software pinpoints the correct line number of the error (‘yes’ – green), points toward the correct area of code, usually by naming the task which contains the error (‘close’ – yellow), or provides an inaccurate or no indication of the location of the error (‘no’ – red).

Error Description. Whether the software provides a clear and helpful description of the error (‘yes’ – green), a correct description which is unclear or confusing (‘close’ – yellow), or no or incorrect error description (‘no’ – red).

The results are reported in Fig. 1 for the individual domains, and in Fig. 2 with an overview. We also provide all data collected (including the actual output messages of the tested software) in a Zenodo repository [11]. We can report that none of the software tested addressed any of our potential semantics errors, with the exception of the VSCode plugin diagnosing the unused predicate error.

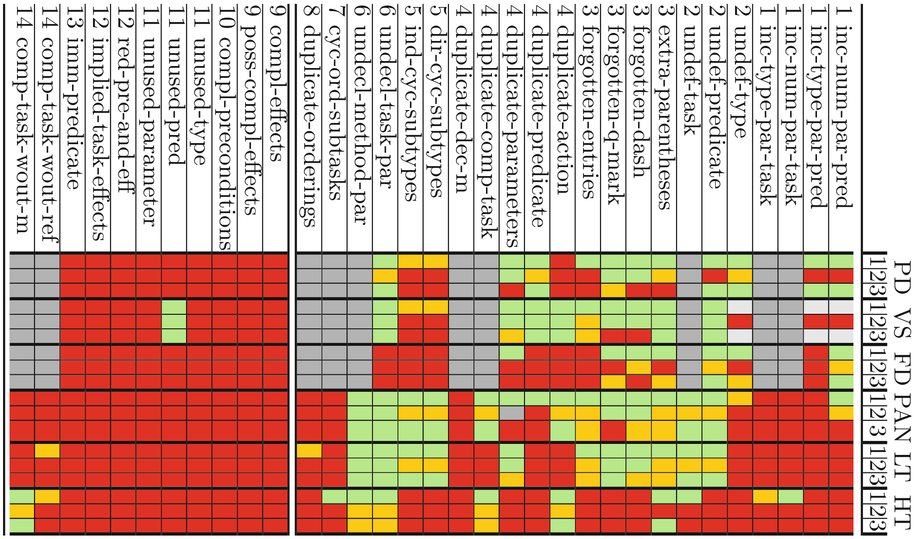


Fig. 1. Results of each software: Planning.Domains (PD), VSCode plugin (VS), Fast Downward (FD), (PAN) DA, Lilotane (LT), Hypertension (HT). We tested error detection (1), line pinpointing (2), and error message quality (3). We first list syntax errors, then (potential) semantic errors. For VS, the lighter shade of green corresponds to errors caught by PD, which the plugin uses as default planner. (Color figure online)

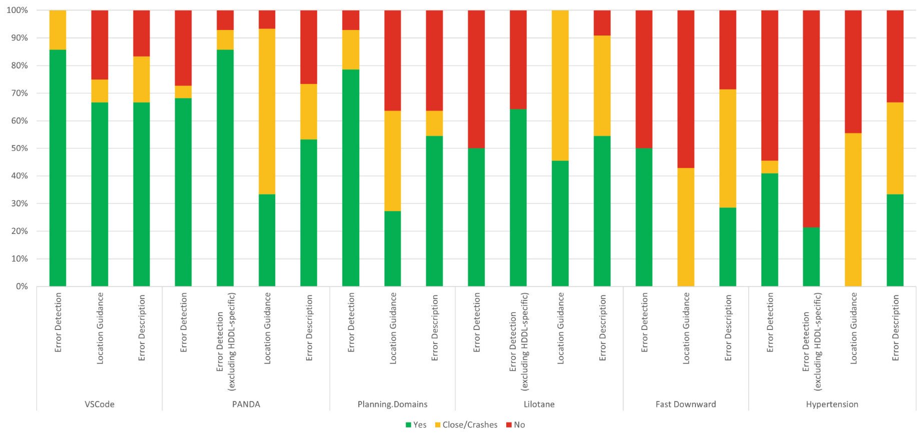


Fig. 2. The overall success rates of the evaluated software. The location guidance and error description rates are percentages of the number of errors caught by the parser, not the total number of errors tested (e.g. if a parser catches 6 of 10 errors, and provides a helpful error message for 3 of them, its success rate for Error Description would be 50%). For hierarchical planners, performance on only the domains which were tested on both classical and hierarchical planning systems is included (called 'excluding HDDL-specific') to allow for fair comparison between the two kinds.

5 Conclusion

We provided a comprehensive list of potential domain modelling errors for classical and hierarchical AI planning. It is accompanied by example domains containing each of these errors, proposed to form the foundation of a set of standardized tests for domain modelling assistance software and improving existing and future PDDL and HDDL parsers.

In our empirical evaluation, we show that a selection of successful well-known – and thus often used – AI planning systems and modeling tools for both PDDL and HDDL domains fail to recognize many of these errors. We thus hope that our list and benchmark set will act as a valuable contribution towards improving these and future software. We furthermore hope that other domain modelers see the benefit in our list and these test cases and thus provide additional benchmarks themselves.

Acknowledgements. We would like to thank Bernd Schattenberg for discussions (and one of the reported errors).

References

1. Bercher, P., Alford, R., Höller, D.: A survey on hierarchical planning - one abstract idea, many concrete realizations. In: Proceedings of the 28th International Joint Conference on AI (IJCAI), pp. 6267–6275. IJCAI (2019)

2. Dolejsi, J.: PDDL visualstudio plugin (2017). https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl
3. Haslum, P., Lipovetzky, N., Magazzeni, D., Muise, C.: An Introduction to the Planning Domain Definition Language. Morgan & Claypool (2019)
4. Helmert, M.: The fast downward planning system. J. Artif. Intell. Res. (JAIR) **26**, 191–246 (2006)
5. Höller, D., Behnke, G., Bercher, P., Biundo, S.: The PANDA framework for hierarchical planning. Künstl. Intell. **35**(3), 391–396 (2021)
6. Höller, D., et al.: HDDL: an extension to PDDL for expressing hierarchical planning problems. In: Proceedings of the 34th AAAI Conference on AI (AAAI), pp. 9883–9891. AAAI Press (2020)
7. Lin, S., Grastien, A., Bercher, P.: Towards automated modeling assistance: an efficient approach for repairing flawed planning domains. In: Proceedings of the 37th AAAI Conference on AI (AAAI), pp. 12022–12031. AAAI Press (2023)
8. Magnaguagno, M.C., Meneguzzi, F., Silva, L.D.: HyperTensioN: a three-stage compiler for planning. In: Proceedings of the 10th International Planning Competition: Planner and Domain Abstracts - Hierarchical Task Network (HTN) Planning Track (IPC 2020), pp. 5–8 (2021)
9. Muise, C.: Planning.Domains. In: ICAPS - Demo 2016 (2016)
10. Schreiber, D.: Lilotane: a lifted SAT-based approach to hierarchical planning. J. Artif. Intell. Res. (JAIR) **70**, 1117–1181 (2021)
11. Sleath, K., Bercher, P.: Experimental results for the PRICAI 2023 paper "Detecting AI planning modelling mistakes - potential errors and benchmark domains" (2023). https://doi.org/10.5281/zenodo.8249690

Responsible AI/Explainable AI

Decision Tree Clustering for Time Series Data: An Approach for Enhanced Interpretability and Efficiency

Masaki Higashi(✉), Minje Sung, Daiki Yamane, Kenta Inamuro, Shota Nagai, Ken Kobayashi, and Kazuhide Nakata

School of Engineering, Tokyo Institute of Technology, Tokyo, Japan
higashi.m.ac@m.titech.ac.jp

Abstract. Clustering is one of the unsupervised learning methods for grouping similar data samples. While clustering has been used in a wide range, traditional clustering methods cannot provide clear interpretations of the resulting clusters. This has led to an increasing interest in interpretable clustering methods, which are mainly based on decision trees. However, the existing interpretable clustering methods are typically designed for tabular data and struggle when applied to time series data due to its complex nature. In this paper, we propose a novel interpretable time-series clustering method with decision trees. To address the interpretability challenges in time-series data, our method employs two separate feature sets, intuitive features for decision tree branching and original time-series observed values for evaluating a given clustering metric. This dual use enables us to construct interpretable clustering trees for time series data. In addition, to handle datasets with a large number of samples, we propose a new metric for evaluating clustering quality, called the *surrogate silhouette coefficient*, and present a heuristic algorithm for constructing a decision tree based on the metric. We show that the computational complexity for evaluating the proposed metric is much less than the silhouette coefficient, which is commonly used in decision tree-based clustering. Our numerical experiments demonstrated that our method constructed decision trees faster than the existing methods based on the silhouette coefficient while maintaining clustering quality. In addition, we applied our method to a time-series data on an e-commerce platform and succeeded in constructing an insightful decision tree.

Keywords: Interpretable Clustering · Timeseries Clustering · Silhouette Coefficient

1 Introduction

Clustering is one of the unsupervised learning methods that groups a set of samples on the basis of their similarities. It is an effective method for understanding the underlying structure of data, and it is applied in various fields

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 457–468, 2024.
https://doi.org/10.1007/978-981-99-7022-3_42

such as marketing [12,17], finance [14,16], natural language processing [5,22], and image recognition [4,6]. Due to its practicality, various kinds of clustering algorithms, such as k-means [15] and hierarchical clustering [11], have been proposed. However, traditional clustering algorithms merely output the grouping of samples and do not provide explanations or interpretations for the resulting groups. Consequently, it is sometimes difficult for these algorithms to ensure the reliability and transparency of the output [20,21].

Against this background, interpretable clustering has attracted much attention in recent years [21]. One of the interpretable clustering methods proposed so far is decision tree clustering. [1–3,8–10,13]. The decision tree [18] is a binary tree model that recursively divides samples into two groups at each node according to certain rules. Typically, at each node, samples are divided into two child nodes on the basis of a threshold value of one feature. Since decision trees have the advantage that their decision rules can be easily visualized as a tree structure, Bertsimas et al. [2] proposed a method that constructs a decision tree for clustering by minimizing the silhouette coefficient [19], which is a clustering evaluation metric, with mixed-integer optimization.

Most of the existing interpretable clustering methods are focused on tabular data. Applying these methods to time series data presents several challenges. In tabular data, each feature often corresponds to a distinct attribute with a specific meaning. Thus, if we construct a decision tree to cluster tabular data based on the threshold values of these features, the resulting clustering rules can be easily interpreted. On the other hand, time series data consists of a sequence of values collected at different time points, and these values often have complex time-dependent relationships. Therefore, if we simply construct a decision tree using the values at each period as features, it becomes difficult to interpret the resulting clustering rules. Moreover, in many business settings, analysis is often performed using data with large sample size. Since the silhouette coefficient used by Bertsimas et al. [2] requires $O(N^2)$ computation for clustering using N samples, the existing method will be computationally expensive for large N.

In this paper, we propose a specialized interpretable time-series clustering algorithm based on decision trees. To the best of our knowledge, this paper is the first to propose a decision tree based interpretable clustering method for time series data. The key to our approach is the use of new interpretable features for the branching rules of the decision tree instead of the observed values. Specifically, while we optimize a certain metric for clustering that is calculated from the observed values, we use the other interpretable features for the branching rules in the decision tree. This approach allows us to construct a decision tree taking into account the similarity of time-series data while keeping the interpretability of the resulting decision tree. Furthermore, to accommodate large amounts of data, we propose a surrogate measure that requires less computation than the silhouette coefficient, called the *surrogate silhouette coefficient*. As a result, our proposed metric reduces the computational complexity from $O(N^2)$ to $O(N)$ for clustering using N data, demonstrating that it allows for more effective use of decision rules generated by decision trees for decision support. It is also shown that the decision rules generated by the decision tree can be used more effectively for decision support.

2 Related Work

Traditional clustering methods such as k-means and hierarchical clustering have a significant drawback: they sometimes fail to provide meaningful results that can be easily understood by humans. As a result, the idea of interpretable clustering has attracted attention. Interpretable models tackle this limitation by generating outcomes that are reliable and straightforward to comprehend, enabling efficient debugging. These models play a crucial role in guiding human decision-making and establishing trust between humans and machines.

To the best of our knowledge, Luc et al. [8] are the first to propose a decision tree-based interpretable clustering method. The proposed algorithm explores a wide range of potential features and cutting points, evaluating the distance between the prototypes of two clusters. Ultimately, it selects the feature and cutting point that yield the greatest distance. This algorithm establishes a fundamental framework for decision tree-based clustering approaches.

Moshkovitz et al. [7] proposed a method that combines decision trees with k-means and k-medians clustering to improve interpretability. By partitioning the data set into clusters using a decision tree, their method can provide a straightforward characterization of each cluster. While the approach provides explanations for clustering results, there are problems with interpretability when applied to time series data, and the number of clusters must be set manually.

Bertsimas et al. [2] proposed an optimization-driven approach to generating interpretable tree-based clustering models. Their algorithm performs clustering by optimizing the silhouette coefficient, which is a measure of clustering quality. Their approach achieves comparable or superior performance to other clustering methods on both synthetic and real-world datasets while offering significantly higher interpretability. However, calculation of silhouette coefficients is computationally expensive, and it takes time to train the decision tree.

3 Method

3.1 Branch Features and Time Series Features

In general tree-based clustering, the same features are used for both constructing decision trees and calculating evaluation metrics such as the silhouette coefficient. This can sometimes make intuitive interpretation difficult. Particularly, when applying decision tree-based clustering to time series data, this problem is likely to be encountered. For example, when clustering a series of sales data of each product on an e-commerce site, the rules of the decision tree may be something like "sold more than 1000 units in April 2020, and moreover, sold more than 1200 units in November 2021." Interpreting what such rules indicate can be challenging, making it difficult to utilize them in decision-making. Therefore, we propose an interpretable time-series data clustering algorithm that uses easily interpretable features for constructing decision trees, while utilizing time series data for calculating evaluation metrics.

Branch Features. Even if decision tree clustering is applied to time-series data, it only determines the features and thresholds of the time-series data for each cluster, making it difficult to link them directly to decision-making. Therefore, to connect the interpretation of clustering using decision tree rules to decision-making, we generate features that are easily understandable to humans and facilitate intuitive comprehension. For example, on an e-commerce site, we can use features such as the growth rate due to a promotion. By incorporating these features, it becomes easier to intuitively understand the clustering results and obtain interpretable insights for decision-making. This approach not only makes the clustering results easier to interpret but also allows flexibility in including different features in the decision tree. This means we can incorporate external data or information not present in the time series data, enabling us to perform clustering and interpretation using additional information. We call these features "branch features" and use them to construct decision tree rules.

Time Series Features. In many supervised decision tree classification models, the focus is on a single feature, and splitting is performed using metrics such as the Gini coefficient. Therefore, it is not a problem to have various types of features mixed together. On the other hand, in the case of decision tree-based clustering, it is an unsupervised learning method where the use of the Gini coefficient is not applicable. Instead of the Gini coefficient, the silhouette coefficient is used. However, it is not desirable to use different types of features since it involves calculating distances on the basis of features. Furthermore, it becomes difficult to capture the meaning of sequential data. Therefore, we refer to time series data as "time series features" and use them to calculate the evaluation metric. We will discuss the evaluation metric in more detail in Sect. 3.2. By using interpretable features for decision tree construction and simultaneously utilizing time series data for calculating evaluation metrics, it becomes possible to perform clustering of time series data that is easily interpretable.

3.2 Surrogate Silhouette Coefficient

When performing clustering, it is necessary to determine a metric for the "goodness" of a cluster. For example, the sum of squares, as used in k-means, considers a clustering to be good when the distance between data points within a cluster and the centroid of that cluster is small. In other words, it is considered good when the data belonging to the same cluster are densely populated. However, the sum of squares, like k-means, can only look at the variance within clusters and cannot observe the spread between clusters. Therefore, in this study, we focus on the silhouette coefficient, which is capable of considering both factors. The silhouette coefficient is a metric for measuring the goodness of clustering, proposed by Rousseeuw [19]. In the case where the number of data points is N, the silhouette coefficient is given as follows:

$$s = \frac{1}{N}\left(\frac{b_1 - a_1}{\max(a_1, b_1)} + \frac{b_2 - a_2}{\max(a_2, b_2)} + \cdots + \frac{b_N - a_N}{\max(a_N, b_N)}\right), \tag{1}$$

where a_i and b_i are referred to as the mean intra-cluster distance and the mean nearest-cluster distance of data i, respectively, and are defined as follows:

$$a_i = \frac{1}{|C_i| - 1}\sum_{j \in C_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \quad b_i = \min_{C_k \in C \setminus \{C_i\}} \frac{1}{|C_k|}\sum_{j \in C_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \tag{2}$$

where C represents the set of all clusters of the entire data, $C_i \in C$ is the cluster containing data i, and $\boldsymbol{x}_i$ represents the features of data i. A silhouette coefficient value close to 1 indicates a good partition, while a value close to -1 indicates a poor partition. The following issue can be considered when calculating the silhouette coefficient.

Issue 1. Due to the use of the L2 norm for distance calculation, each term of the sum in (2) cannot be combined.

Issue 2. It is necessary to calculate the average distance between all clusters, which results in a large number of clusters for the calculation of b_i.

Issue 3. The computational complexity is high because the denominator $\max(a_i, b_i)$ varies between data points.

To address these issues, we propose a surrogate silhouette coefficient that requires less computational effort than the silhouette coefficient.

Solution to Issue 1. To address the first issue, we use the squared L2 norm when calculating the distance between data points, allowing us to group common terms together. In this case, for data point i belonging to cluster $C_i \in C$, the mean intra-cluster distance a_i and the mean nearest-cluster distance b_i are as follows:

$$a_i = \frac{1}{|C_i| - 1}\sum_{j \in C_i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, \quad b_i = \min_{C_k \in C \setminus \{C_i\}} \frac{1}{|C_k|}\sum_{j \in C_k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2. \tag{3}$$

Solution to Issue 2. To address the second issue, we propose calculating the average distance for the two clusters split within a branch node. Let us denote the clusters within the branch node as C_1, C_2 and suppose data i belongs to cluster C_1. Then, the mean intra-cluster distance a_i and the mean nearest-cluster distance b_i are as follows:

$$a_i = \frac{1}{|C_1| - 1}\sum_{j \in C_1} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \frac{1}{|C_1| - 1}\left(|C_1| \|\boldsymbol{x}_i\|^2 - 2\boldsymbol{x}_i^T \sum_{j \in C_1} \boldsymbol{x}_j + \sum_{j \in C_1} \|\boldsymbol{x}_j\|^2\right) \tag{4}$$

$$b_i = \frac{1}{|C_2|}\sum_{j \in C_2} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \frac{1}{|C_2|}\left(|C_2| \|\boldsymbol{x}_i\|^2 - 2\boldsymbol{x}_i^T \sum_{j \in C_2} \boldsymbol{x}_j + \sum_{j \in C_2} \|\boldsymbol{x}_j\|^2\right). \tag{5}$$

Solution to Issue 3. To address the last issue, we calculate the mean intra-cluster distance a and the mean nearest-cluster distance b for each cluster. For the clusters $C_k, k \in \{1, 2\}$ that have been split within a branch node, a and b are as follows:

$$\begin{aligned} a_{C_k} &= \frac{1}{|C_k|}\sum_{i\in C_k} a_i = \frac{1}{|C_k|\left(|C_k|-1\right)}\sum_{i\in C_k}\left(|C_k|\left\|\boldsymbol{x}_i\right\|^2 - 2\boldsymbol{x}_i^T\sum_{j\in C_k}\boldsymbol{x}_j + \sum_{j\in C_k}\left\|\boldsymbol{x}_j\right\|^2\right) \\ &= \frac{2}{|C_k|-1}\left(\sum_{j\in C_k}\left\|\boldsymbol{x}_j\right\|^2 - |C_k|\left\|\sum_{j\in C_k}\frac{\boldsymbol{x}_j}{|C_k|}\right\|^2\right) \end{aligned} \tag{6}$$

$$\begin{aligned} b_{C_k} &= \frac{1}{|C_k|}\sum_{i\in C_k} b_i = \frac{1}{|C_k|\left|C_{\tilde{k}}\right|}\sum_{i\in C_k}\left(\left|C_{\tilde{k}}\right|\left\|\boldsymbol{x}_i\right\|^2 - 2\boldsymbol{x}_i^T\sum_{j\in C_{\tilde{k}}}\boldsymbol{x}_j + \sum_{j\in C_{\tilde{k}}}\left\|\boldsymbol{x}_j\right\|^2\right) \\ &= \sum_{l\in\{1,2\}}\left(\frac{1}{|C_l|}\sum_{j\in C_l}\left\|\boldsymbol{x}_j\right\|^2\right) - 2\left(\sum_{j\in C_k}\frac{\boldsymbol{x}_j}{|C_k|}\right)^T\left(\sum_{j\in C_{\tilde{k}}}\frac{\boldsymbol{x}_j}{\left|C_{\tilde{k}}\right|}\right), \end{aligned} \tag{7}$$

where if $k = 1$ then $\tilde{k} = 2$, and if $k = 2$ then $\tilde{k} = 1$. Using (6) and (7), the surrogate silhouette coefficient can be calculated as follows:

$$s = \frac{|C_1|}{|C_1| + |C_2|}\frac{b_{C_1} - a_{C_1}}{\max\left(a_{C_1}, b_{C_1}\right)} + \frac{|C_2|}{|C_1| + |C_2|}\frac{b_{C_2} - a_{C_2}}{\max\left(a_{C_2}, b_{C_2}\right)}. \tag{8}$$

Thus, the computational complexity has been reduced from $O(N^2)$ to $O(|C_1| + |C_2|)$ using the surrogate silhouette coefficient.

3.3 Algorithm for Decision Tree Construction

The algorithm for constructing a decision tree using Sect. 3.1 is as follows.

Step1: Split the data within a branch node into two subsets, C_1 and C_2, using a threshold value based on the branch feature values of the data points in the node. Calculate the evaluation metric using the time series features of C_1 and C_2. If the number of data points in either C_1 or C_2 is less than $N_{\min}$, set the evaluation metric value to $-\infty$. Repeat the above steps for all data points within the branch node and all features to calculate the maximum evaluation metric value, denoted as max-score. The corresponding partitioned data subsets are denoted as C_1^* and C_2^*. If max-score is $-\infty$, proceed to Step 3. Otherwise, proceed to Step 2.

Step2: If max-score is not $-\infty$, calculate the silhouette coefficient using the previously partitioned data subsets and the newly partitioned data subsets, C_1^* and C_2^*. If the silhouette coefficient value is less than a threshold value, Th, proceed to Step 3. If the silhouette coefficient value is greater than or equal to Th, split this branch node into C_1^* and C_2^*, creating new branch nodes as child nodes in the lower level. Then proceed to Step 4.

Step3: Make this branch node a leaf node without splitting it into C_1^* and C_2^*. Then proceed to Step 4.

Step4: Move to the right adjacent branch node and restart from Step 1. If there is no branch node on the right side, move to the leftmost node in the lower level and restart from Step 1. If there is no such node either, terminate the algorithm.

$N_{\min}$ and Th are hyperparameters. $N_{\min}$ represents the minimum number of data points within a cluster, and Th is the minimum threshold for the silhouette coefficient.

4 Numerical Experiments

To investigate the effectiveness and practicality of our method, we conducted numerical experiments with synthetic and real datasets[1].

4.1 Comparison of Computational Time

We first compare the computational time and clustering accuracy of two methods: a baseline method that uses the silhouette coefficient as the objective function and our proposed method that uses the surrogate silhouette coefficient as the objective function. Furthermore, we were concerned that the number of divisions might change in the algorithm described in Sect. 3.3, so we made a modification to accurately compare computational times. Specifically, the modification was as follows: the condition for transitioning to Step 3 in Step 2 was changed to a condition under which "the silhouette coefficient computed on the newly divided data groups C_1^* is smaller than the silhouette coefficient based on the data groups C_2^* that had been divided before the new division."

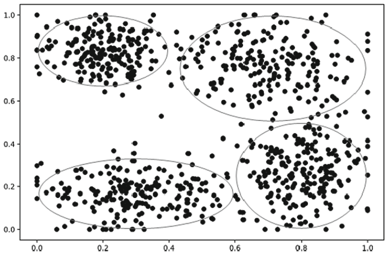


Fig. 1. Scatter plots of synthetic data consisting of four clusters.

[1] All code and scripts for our method are available at https://github.com/tokyotech-nakatalab/interpretable_time-series_clustering.

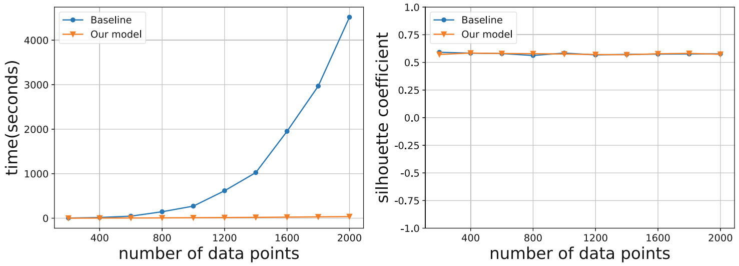


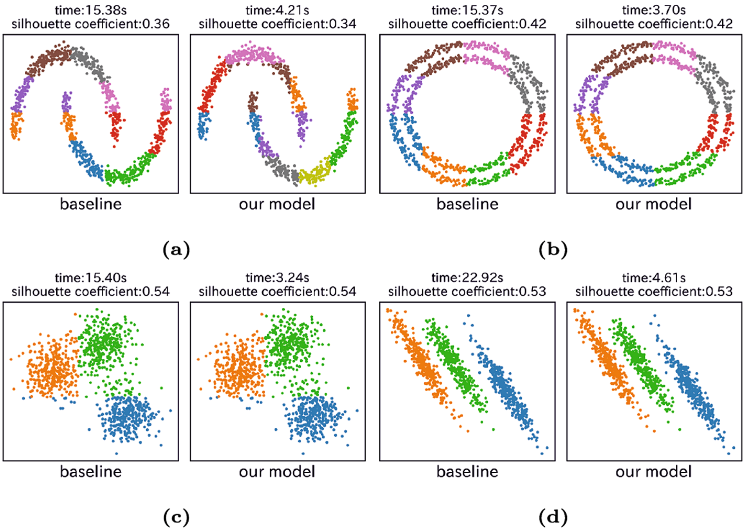
Fig. 2. Computational time and silhouette coefficient for each number of data points.

In this experiment, we used two-dimensional synthetic data. The datasets consist of four clusters, and each data point is generated from its corresponding normal distribution. We show an illustration of the synthetic dataset in Fig. 1. We changed the number of data points from 400 to 4,000 in increments of 400 and measured the computational time and clustering accuracy for each case. The clustering accuracy was evaluated with the silhouette coefficient. The results of the computational time and clustering accuracy are shown in Fig. 2. For the baseline method, the computational time significantly increased as the number of data points grows. In contrast, the proposed method showed only a slight change in computational time. When the number of data points was 4,000, the computational time of the baseline method was 4,562 s, whereas the proposed method took only 36 s. This means the computational time of the surrogate silhouette coefficient was reduced to 1/125 compared with the silhouette coefficient. Additionally, both methods achieved a clustering accuracy with a silhouette coefficient of approximately 0.57, indicating little difference in clustering performance between the two methods.

From the results of the experiments, it was observed that the surrogate silhouette coefficient significantly reduced the computational time without sacrificing clustering accuracy compared with the original silhouette coefficient in Fig. 1.

4.2 Comparison Based on Distributions

From Sect. 4.1, it was observed that the surrogate silhouette coefficient significantly reduced the computational time without sacrificing clustering accuracy. In this section, we examine the limitations of the silhouette coefficient, especially our proposed one, and the usefulness of branch features by using data with various distributions. In Fig. 3, both methods construct a decision tree using the features in Sect. 3.1 and the algorithm in Sect. 3.3, but the baseline uses the silhouette coefficient, and our model uses the surrogate silhouette coefficient as an evaluation metric. First, for non-convex clustering cases where scoring based on the silhouette coefficient is not suitable, it can be seen that the clustering results were not successful, as shown in Figs. 3a and 3b. Moreover, since the surrogate silhouette coefficient evaluates the data within the branch nodes of the decision tree, rather than considering the evaluation across the entire dataset, the results



(a) **(b)** **(c)** **(d)**

Fig. 3. Clustering results for various distributions

were heavily influenced by the previous splits. Therefore, while the clustering based on the silhouette coefficient in Fig. 3a attempted to create cohesive clusters, the surrogate silhouette coefficient generated fragmented clusters, as seen in the case of the brown colored cluster. In Fig. 3d, clustering was performed by incorporating $1.65x + y$ as one of the branch features for each data point (x, y). This suggests that with the addition of branch features, it may be possible to solve the problem of decision tree clustering that cannot create divisions that are not parallel to the axes.

4.3 Application for Time Series Data

We investigate the practicality of our method with a real dataset of purchase histories from an e-commerce marketplace provided by Rakuten Group, Inc. We clustered genres of products with their weekly sales quantity from 2019 to 2020. For each genre, We used the weekly sales quantity for the time series feature, where each feature is standardized so that its mean and variance are 0 and 1, respectively. For branch features, we used three features: 1) the proportion of sales quantity for each season, 2) the sales growth rates compared with the corresponding period of the previous year, and 3) the sales growth rates compared with the previous week.

Figure 4 shows a decision tree constructed using only time series data on the left and another constructed using the proposed method on the right. By using branch features, it becomes easier to interpret the characteristics of each cluster. This enables us to devise business strategies tailored to each cluster's unique features. The silhouette coefficient value for the decision tree with the branch features was 0.019.

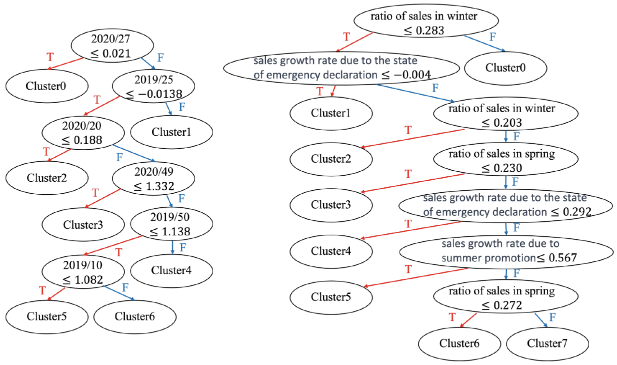


Fig. 4. Decision tree constructed using only time series data (left), decision tree constructed using branch features (right).

4.4 Accuracy of Time Series Prediction Using Clustering Results

To assess the validity of the clustering results, we examined whether the clustering has yielded meaningful outcomes related to the problem it aims to solve from various perspectives. We compared evaluation metrics of time series prediction models for sales data with and without cluster labels as features.

Experimental Setup. We evaluated the performance of the sales prediction model using the data from Sect. 4.3. We first normalized the sales data for each genre to the range of 0–1 and predicted weekly sales for each genre on the basis of the sales of the previous five weeks. We also used the presence of promotions as a feature. The data was divided by holding out the 2020 data as the test set. The 2019 data was divided into training and validation sets, using time-series K-fold cross-validation, and the prediction model was trained using LightGBM. In addition, for generating cluster labels, we performed clustering using the proposed method. The normalized sales quantity was used as a time-series feature, and as branch features, we utilized the rate of increase in sales quantity comparing the week of and two weeks immediately prior to a promotion.

Results. The evaluation metric for the prediction model was RMSE. Additionally, we used k-means with Dynamic Time Warping (DTW) as a baseline for the clustering methods. According to Table 1, adding cluster labels as features resulted in a lower RMSE. Furthermore, even among those that added cluster labels as features, our model achieved a lower RMSE than the combination of DTW and k-means. Such results suggest that the cluster labels generated from our model can be useful for sales prediction.

In addition, the decrease in RMSE due to the addition of cluster labels in Setting 2 was most pronounced in the samples during promotions. For the case where the decrease in RMSE was sorted in descending order for each sample, the

Table 1. Average and standard deviation of RMSE for 20 trials in each setting

Without Cluster Label	With Cluster label	
	DTW+kmeans	our method
0.2282 ± 0.0375	0.2144 ± 0.0126	**0.2042** ± 0.0077

Table 2. Proportion of samples during promotions for each Top@K

Top@100	Top@500	Top@1000
100.0%	64.6%	41.9%

proportion of sales weeks included in the Top@K is shown in Table 2. Considering that the overall proportion was 11.5%, this value indicates that the cluster labels have captured the trend of promotions for each cluster.

5 Conclusion

We proposed an interpretable and fast time-series clustering method based on decision trees. To ensure interpretability of time series clustering, we used interpretable features for branching rule in decision tree and time-series features for evaluating the clustering quality. To speed up the computation for constructing decision tree, we also proposed to construct a decision tree with a new metric, called *surrogate silhouette coefficient*. In our experiments, our method can significantly reduced the computation time while keeping clustering quality. Also, our method succeeded in yielding an interpretable and reasonable decision tree with a real dataset.

For future work, since our current method requires us to design "branch features" by hand, it is interesting to explore how to generate such features automatically. Also, a user study is required to evaluate the interpretability of our method quantitatively and qualitatively.

Acknowledgments. This study was conducted as a part of the Data Analysis Competition hosted by Joint Association Study Group of Management Science. The authors would like to thank the organizers and Rakuten Group, Inc. for providing us with a real data set.

References

1. Basak, J., Krishnapuram, R.: Interpretable hierarchical clustering by constructing an unsupervised decision tree. IEEE Trans. Knowl. Data Eng. **17**, 121–132 (2005)
2. Bertsimas, D., Orfanoudaki, A., Wiberg, H.: Interpretable clustering: an optimization approach. Mach. Learn. **110**, 89–138 (2021)
3. Blockeel, H., De Raedt, L., Ramon, J.: Top-down induction of clustering trees. In: Proceedings of the 15th International Conference on Machine Learning, pp. 55–63 (1998)

4. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision, pp. 132–149 (2018)
5. Chang, E., Shen, X., Yeh, H-S., Demberg, V.: On training instance selection for few-shot neural text generation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9707–9718 (2021)
6. Coleman, G.B., Andrews, H.C.: Image segmentation by clustering. Proc. IEEE **67**, 773–785 (1979)
7. Dasgupta, S., Frost, N., Moshkovitz, M., Rashtchian, C.: Explainable k-means and k-medians clustering. In: Proceedings of the 37th International Conference on Machine Learning, pp. 7055–7065 (2020)
8. De Raedt, L., Blockeel, H.: Using logical decision trees for clustering. In: International Conference on Inductive Logic Programming, pp. 133–140 (1997)
9. Fraiman, R., Ghattas, B., Svarc, M.: Interpretable clustering using unsupervised binary trees. Adv. Data Anal. Classif. **7**, 125–145 (2013)
10. Ghattas, B., Michel, P., Boyer, L.: Clustering nominal data using unsupervised binary decision trees: comparisons with the state of the art methods. Pattern Recogn. **67**, 177–185 (2017)
11. Joe, J., Ward, J., Jr.: Hierarchical grouping to optimize an objective function. Am. Stat. Assoc. **58**, 236–244 (1963)
12. Kim, K., Ahn, H.: A recommender system using GA k-means clustering in an online shopping market. Expert Syst. Appl. **34**, 1200–1209 (2008)
13. Liu, B., Yiyuan, X., Philip, S, Y.: Clustering through decision tree construction. In: Proceedings of the Ninth Conference on Information and Knowledge Management, pp. 20–29 (2000)
14. Lux, T., Marchesi, M.: Volatility clustering in financial markets: a micro-simulation of interacting agents. Int. J. Theor. Appl. Finan. **3**, 675–702 (2000)
15. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
16. Onnela, J.-P., Kaski, K., Kertész, J.: Clustering and information in correlation based financial networks. Eur. Phys. J. B **38**(2), 353–362 (2004). https://doi.org/10.1140/epjb/e2004-00128-7
17. Punj, G., Stewart, D.W.: Cluster analysis in marketing research: review and suggestions for application. J. Mark. Res. **20**(2), 134–148 (1983)
18. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
19. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
20. Saisubramanian, S., Galhotra, S., Zilberstein, S.: Balancing the tradeoff between clustering value and interpretability. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 351–357 (2020)
21. Yang, H., Jiao, L., Pan, Q.: A survey on interpretable clustering. In: Proceedings of 40th Chinese Control Conference, pp. 7384–7388 (2021)
22. Yoon, S., Dernoncourt, F., Kim, D., Bui, T., Jung, K.: A compare-aggregate model with latent clustering for answer selection. In: Proceedings of the 28th International Conference on Information and Knowledge Management, pp. 2093–2096 (2019)

The Ethical Evaluation Method of Algorithmic Behavior Based on Computational Experiments

Fangyi Chen[1], Xiao Xue[1](✉), and Xiao Wang[2]

[1] Tianjin University, Tianjin 300354, China
[2] University of Chinese Academy of Sciences, Beijing 100190, China
{chenfangyi,jzxuexiao}@tju.edu.cn, x.wang@ia.ac.cn

Abstract. The widespread application of artificial intelligence algorithms has brought about various ethical concerns, such as algorithm discrimination. While there have been some efforts focused on enhancing the ethical performance of algorithms, the evaluation of their ethical behavior has been largely neglected. However, conducting practical evaluations is often infeasible due to factors such as cost, legal, and other constraints. In this context, computational experiments have emerged as novel and powerful computational theories and tools for quantitative analysis in complex social systems. This paper proposes an ethical evaluation method of algorithmic behavior, called EMAB, which leverages the computational experiment and simulation to construct an AI-driven artificial society, providing a dynamic and feedback-based environment for evaluating the ethics of algorithms. EMAB includes users, algorithms, and a dynamic data circulation mechanism between them. Taking the recommendation algorithm as an example, we design test scenarios to verify the superiority of the fair recommendation algorithm over the unfair recommendation algorithm. The experimental results illustrate the effectiveness and necessity of EMAB. The proposed method provides a novel perspective for algorithm evaluation involving ethics.

Keywords: Algorithm evaluation · Agent-based Modeling · Computational experiments · AI ethics

1 Introduction

With the advent of the Internet and big data, artificial intelligence (AI) algorithms have permeated into human daily life and even the realm of national governance, transforming the functioning and progress of society comprehensively. While algorithms have empowered society, their ethical issues have also triggered a series of debates and research. Issues such as information cocoon rooms [1], algorithm black boxes [2] and algorithm discrimination [3] continue to arise. In this context, algorithmic governance has become an increasingly concerning topic for researchers [4].

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 469–480, 2024.
https://doi.org/10.1007/978-981-99-7022-3_43

Most research on algorithm ethics focused on improving algorithmic ethical performance and developing governance frameworks based on legal norms. Few researchers have paid attention to evaluating the ethical behavior of algorithms, which is essential for ensuring they operate in an ethical and socially responsible manner. Evaluating the ethical behavior of algorithms is a complex task that presents many challenges. Firstly, algorithms are self-evolving, continuously updating their knowledge through interaction and information. Secondly, the complex relationships between users can significantly affect users' behavior and the performance of algorithms that rely on user feedback. Furthermore, privacy concerns and legal restrictions often make it difficult to access sensitive information required to evaluate the ethics of algorithms.

Taking the fairness evaluation as an example, existing methods can be divided into two categories: one is based on indicators derived from static benchmark datasets without feedback [5,6], but static data can hardly reflect the complexity of real social systems. Another uses simulation to study the long-term fairness performance [7]. However, it does not take into account the impact of complex interactions between users. In fact, the interaction between users affects their behavior and preferences, which in turn affects the fairness performance of the algorithm.

The computational experiment provides a computational method for the quantitative analysis. It cultivates the computing laboratory of the real system in the information world, forming the artificial society [8]. This paper proposes a method for evaluating the ethical behavior of algorithms based on computational experiments. Overall, the contributions of this paper are mainly:

- The paper introduces EMAB, a novel ethical evaluation method for algorithmic behavior, which provides a dynamic and feedback simulation environment to evaluate long-term fairness performance.
- The impact of user interaction is explored experimentally. The results highlight the inadequacies of traditional evaluation methods and emphasize the need for more realistic environments.
- A comparative experiment between a fair recommendation algorithm and an unfair recommendation algorithm was conducted to demonstrate the efficacy of EMAB. The results indicate that the fairness of both algorithms becomes worse in the dynamic environment, but the fair algorithm still outperforms the unfair algorithm.

2 Related Work

2.1 Ethical Research on Algorithmic Behavior

Nowadays, issues such as information cocoons narrowing cognition [9], discriminatory pricing harming user rights [10], and unfair judicial rulings [11] have raised concerns about AI algorithms. Some works aim to directly improve the ethical performance of algorithms [12], while others focus on evaluating their ethics. Algorithm auditing [13,14] involves a series of methods for reviewing algorithms to assess whether they comply with laws and ethical considerations. However, challenges such as the lack of uniform standards and difficult access have hindered the implementation. In terms of the fairness evaluation of algorithms, there are some open-source tools that provide a visual representation of the ethical performance [15,16]. However, these methods rely on static datasets and are unable to evaluate algorithms in an interactive feedback environment. The literature introduces ml-fairness-gym [7], which utilizes simulation to investigate the long-term fairness of algorithms. The article demonstrates that the issue of long-term fairness is complicated by dynamic feedback. However, this method does not consider the effects of intricate interactions between users.

2.2 Computational Experiments

Computational experiments have emerged as a new social science research method. It can overcome limitations such as costs, legal restrictions, and ethical considerations. Moreover, this method combines qualitative and quantitative analysis to better study complex phenomena and dynamic evolution. Currently, the computational experiment has become a mainstream analysis method for analyzing complex systems successfully [17–19]. The computational experiment method can provide an artificial laboratory for the ethical evaluation of algorithmic behavior in a virtual environment, thereby solving the constraints of cost, information acquisition, legal norms, and morality in the ethical evaluation of algorithmic behavior [20–23]. Artificial society modeling is a key step in computational experiments. It needs to consider the agent's autonomous decision-making, heterogeneity, bounded rationality, and learning evolution mechanisms [24,25]. However, few works focus on including anthropomorphic features in agent modeling, which can enhance the simulation of human society.

3 The Ethical Evaluation Method of Algorithmic Behavior

3.1 Overall Framework

The EMAB consists of a recommendation algorithm, an artificial society, and a loop feedback mechanism between them. During each recommendation cycle, the algorithm collects user feedback data and generates a recommendation list. The recommendation results are then analyzed using evaluation indicators. The user agent perceives the recommendation list and then scores the items based on the decision-making mechanism, and updates its states. After all users complete the scoring, the algorithm collects the scoring data and performs self-evolution, updating existing knowledge, and making the next recommendation, thus forming a dynamic circular feedback mechanism between the algorithm and users. Figure 1 depicts the overall framework of the proposed method, taking the fairness evaluation as an example.

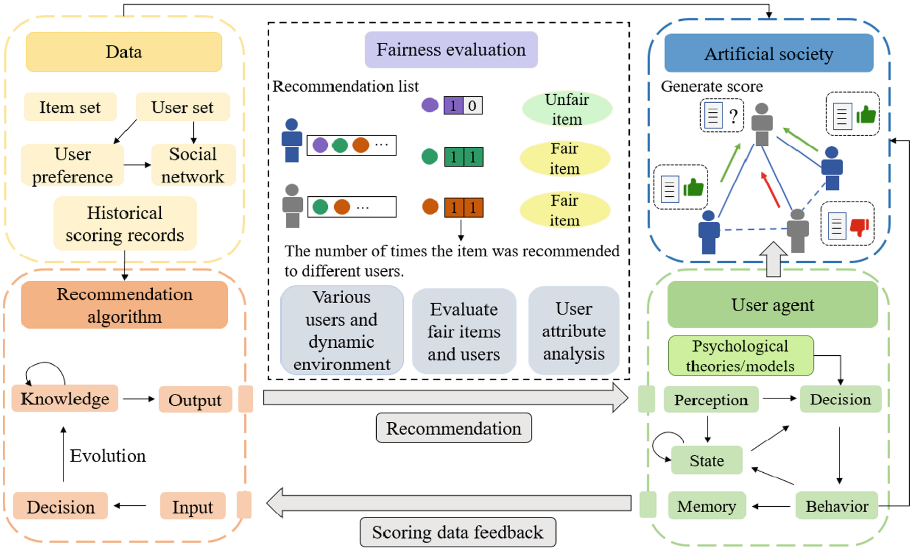


Fig. 1. The framework of the EMAB method.

3.2 Artificial Society

Individual-Level Embedded-Psychology Modeling. The user agent is an autonomous individual with the ability to perceive, make decisions, engage in behavior, and optimize his actions. The formal expression of the user agent is represented in the Eq. (1). R denotes the agent's time-invariant characteristics, while S_t denotes the time-variant ones. E_t encompasses external events observed by the user agent, which modify its state and behavior. Y_t represents the decision-making mechanism used by the agent to produce behaviors in response to external stimuli and interactions. V_t is the set of behaviors, and N represents the bounded rational constraint condition of the agent.

$$Agent =< R, S_t, E_t, Y_t, V_t, N > . \tag{1}$$

The Perception module embodies its ability to perceive external stimuli E_t and output the perceived information Per_t. The State module comprises a set of states, which include R and S_t, as shown in Table 1. The neighbors refer to other users who share a social relationship with the current user and the scoring record denotes the memory capacity of the agent, enabling it to retain historical scoring information within a specified time range. The user makes an action $Act_t \in V_t$ according to Per_t, Y_t, $Memory_t$ and $State_t$ at time t, and its formal representation is shown in Eq. (2). Then, the user updates the state according to Per_t and Act_t.

$$Decision : Per_t \times Y_t \times Memory_t \times State_t \rightarrow Act_t. \tag{2}$$

Table 1. Attributes of the User Agent

Type	Attribute	Attribute description
Fixed	User identity(ID)	Unique user ID
	Gender	User gender
	Age	User age
Mutable	Preference vector	Mean ratings
	Neighbors	Other users with social relationship
	Scoring record	User's historical scoring records

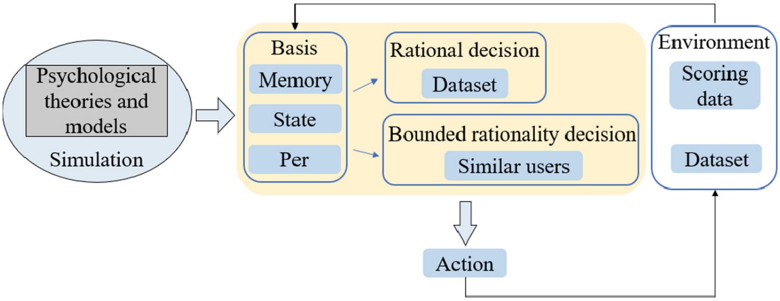


Fig. 2. The decision of psychology-embedded user agents .

Based on the bandwagon effect [26] and crowd psychology [27], we imbue the user agent with anthropomorphic features. Specifically, the user's score for an item will be influenced by other users who have similar preferences. This phenomenon reflects how individuals with similar tastes interact in real life. The process of generating simulated user ratings is shown in Algorithm 1. Figure 2 illustrates the decision-making framework for the psychology-embedded user agent.

Social-Level Modeling. EMAB builds a social network within the artificial society by referring to the literature [28]. Specifically, at time t, the connection probability between the newly added user node u_t and the user node u_w that has joined in the past w time is based on the similarity of their preferences and the influence, as shown in Eq. (3). $\bar{S}_{tw}$ represents the standardized preference similarity. $cos(\cdot)$ represents the cosine similarity, P represents the user's preference vector, and q indicates the number of nodes that have joined the artificial society. $\bar{F}_w$ is the normalized influence of node u_w, represented by the degree d_w of node u_w, as shown in Eq. (6). Once the network is formed, users with connected edges become neighbors.

$$\pi_{tw} = \alpha \bar{S}_{tw} + (1 - \alpha)\bar{F}_w. \tag{3}$$

$$\bar{S}_{tw} = S_{tw}(\sum_{j=0}^{q-1} S_{tj})^{-1}. \tag{4}$$

```
Algorithm 1. User agent rating behavior.
Input: The collection of items to be rated $I_{rate}$, the collection of users $U$, user historical rating data $R$, rating similarity matrix between users $SimRating$, the number of similar users $K$.
Output: User $u$'s rating for item $i$, $u \in U$, $i \in I_{rate}$.
1: Refer to ratings from public datasets.
   If the user $u$ and item $i$ being scored are present in the public data set, utilize the score assigned to the user as the score for the item. If not, go to the next step.
2: Refer to the rating values of $K$ similar users based on the interaction.
   Choose the users who have rated the target item and have a positive rating similarity with the target user. Select the top $K$ users and perform a weighted summation based on their rating similarity, e.g., $r_{ui} = \sum_{j=0}^{K} SimRating_{uj} * r_{ji}$. $r_{ji}$ represents user $j$'s rating on item $i$.
```

$$S_{tj} = cos(P_t, P_j) = \frac{P_t^T P_j}{|P_t||P_j|}. \tag{5}$$

$$\bar{F}_w = d_w(\sum_{j=0}^{q-1} d_j)^{-1}. \tag{6}$$

3.3 Recommendation Algorithms and Metrics

Recommendation algorithms utilize user-item historical interaction data and other information, such as social networks, to capture user preferences and item characteristics, known as knowledge. Algorithm 2 describes the operation process of the recommendation algorithm based on embedding.

The recommendation performance is measured using R, as shown in Eq. (7), where $\mathbb{E}$ means mathematical expectation and $Rec(u)$ means the proportion of items that the user u gives high ratings in the recommendation list. $LikeNum_u$ represents the number of highly rated items by user u. N_u represents the recommendation list for u. $len(\cdot)$ denotes the length of the set.

$$\mathcal{R} = \mathbb{E}_{u \in U}[Rec(u)]. \tag{7}$$

$$Rec(u) = LikeNum_u / len(N_u). \tag{8}$$

Fairness means that the recommendation should be independent of sensitive attributes. For example, if item i is only recommended to male users, it is unfair because it heavily depends on gender. However, for recommendations that are closely related to sensitive attributes (i.e., items that are uniquely specific to one gender or age), blindly guaranteeing fairness may lead to poor recommendations, so a trade-off needs to be made. Here, we consider fairness within general recommendation scenarios. Specifically, we use Eq. (9) to measure the fairness. Using gender as an example, Arr_i is an array that records the number of times item i is recommended to male users and female users respectively. $Var(i)$ is the

Algorithm 2. The recommendation algorithm process.

Input: Collection of users U, collection of items I, user historical rating data R, preference similarity matrix between neighbors $SimPre$.

Output: A collection of recommended items for users.

```
1: Train a recommendation model based on $R$.
   Train the model according to the specific recommendation algorithm.
2: Based on prior knowledge, predict user ratings for items.
   Algorithms use learned knowledge to predict user ratings for items.
3: Generate a recommendation list from items with high predicted scores.
   Recommended items for the current user are selected based on predicted scores greater than the threshold. If no such items exist, proceed to step 4.
4: Refer to the user's social relations.
   Select the user $u'$ with the highest similarity to the current user's preference among the neighbors, as Eq. (5). If the algorithm generates a recommendation list for $u'$, assign it to $u$. If there is no recommendation list for $u'$, search for the second most similar user until a recommendation list is found for $u$. If no suitable recommendation list can be found for $u$ after traversing the neighbors, proceed to step 5.
5: Generate a list of random items as a recommendation list.
   Generate a random recommendation list according to the specified length.
6: Collect user feedback data and update knowledge.
   Collect user feedback scoring data, fine-tune the recommendation model, update the knowledge, and then proceed to step 2 for the next recommendation.
```

variance of Arr_i. A smaller $Var(i)$ indicates that the recommendation for item i is fairer.

$$\mathcal{F} = \mathbb{E}_{i \in I}[Var(i)]. \tag{9}$$

$$Var(i) = \mathbb{E}[|Arr_i - \mathbb{E}[Arr_i]|^2]. \tag{10}$$

In addition, we identify the set of items with $Var(i)$ greater than the threshold β, denoted as $UnfairItem$. These items are not considered to be fairly recommended. If the unfairly recommended item appears in user u's recommendation list more than γ times, we consider the recommendation for u to be unfair, as shown in Eq. (11) and (12).

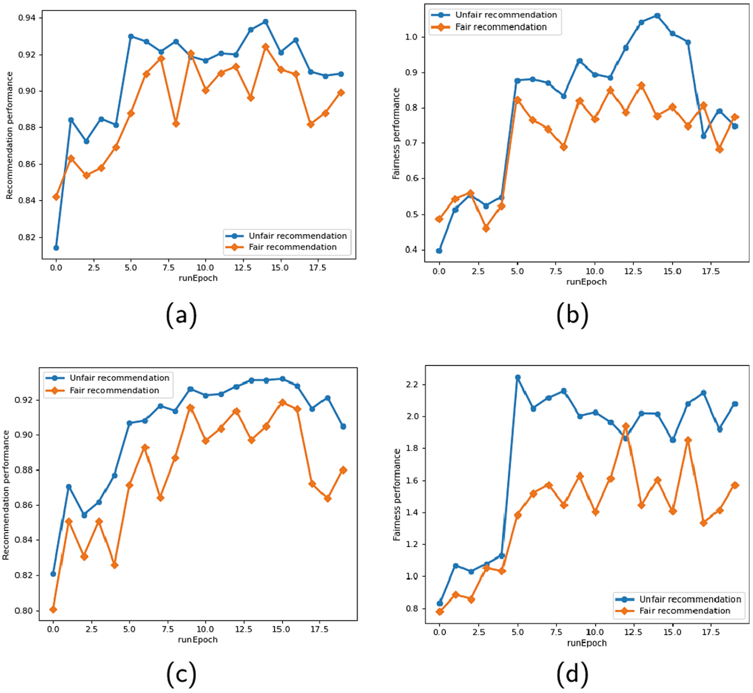
$$UnfairItem = \{i\}, i \in I, Var(i) > \beta. \tag{11}$$

$$UnfairUser = \{u\}, u \in U, len(N_u \cap UnfairItem) > \gamma. \tag{12}$$

4 Experiment

4.1 Experiment Setup

The experiments are based on the MovieLens-1M [29] dataset. Firstly, the fair recommendation algorithm and unfair recommendation algorithm [30] are trained based on the dataset. According to the description in Sect. 3, an artificial society is constructed. The algorithm interacts dynamically with users, and co-evolves based on the data cycle. We evaluate the performance of the algorithm after each round of recommendations. The parameter settings are shown in Table 2. A fair recommendation algorithm learns user representations through adversarial training and self-supervised learning so that they contain less sensitive information. Besides, it restricts the predictive score to be independent of user-sensitive attributes through regularization methods. Unfair algorithms do not consider the impact of sensitive information.



(a)

(b)

(c)

(d)

Fig. 3. Comparison of the fair recommendation algorithm and the unfair recommendation algorithm. The x-axis runEpoch indicates the number of artificial society runs. (a) Recommendation performance under age attribute. (b) Fairness performance under age attribute. (c) Recommendation performance under gender attribute. (d) Fairness performance under gender attribute.

4.2 Analysis of Experimental Results

Comparison of the Fair Recommendation Algorithm and the Unfair Recommendation Algorithm. First, we compare the performance of fair and unfair recommendation algorithms in the artificial society. Figure 3 shows the recommendation and fairness performance of the two algorithms when age and gender are considered sensitive attributes. The experimental results demonstrate that the fairness of the recommendation results decreases when user interaction and the cyclic interaction between the algorithm and users are taken into

Table 2. Experimental parameter settings.

System variable	Value
Number of users	[30,40,...,100,200,...,1000,2000]
Length of recommendation list	20
runEpoch	20
Evaluation cycle	1
The number of fine-tuned training	40
α	0.5
β	1
γ	15
K	[10,15,20,25]

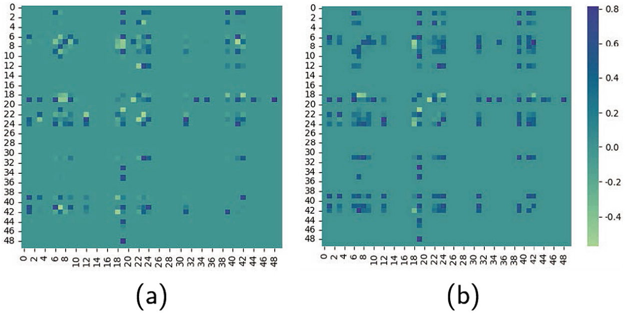


Fig. 4. User preference similarity heat map. (a) The first recommendation. (b) The fifteenth recommendation.

account. This further emphasizes the importance of considering the dynamic environment when evaluating algorithm behavior. However, the fair recommendation algorithm can still achieve better fairness performance without significant loss in recommendation performance, which aligns with our expectations.

User-Centric Analysis. We analyze the change in user preference similarity during the recommendation and plot the heat map of Pearson correlation coefficient [31] of preference vectors between connected users, as shown in Fig. 4, where darker colors indicate greater similarity. It is evident that the similarity of preferences between users gradually increases, which confirms our previous ideas and reveals that this phenomenon can negatively impact the fairness performance of the recommendation algorithm to some extent. Additionally, according to Eq. (12), we find out the users who are recommended unfairly and visualize the situation in the social network, as shown in Fig. 5. Red nodes represent users who have not received fair recommendations, and green nodes represent users who have received fair recommendations. Through computational experiments, the situation of each agent can be analyzed at the micro level.

Exploring the Causes of Unfairness from a Data Perspective. We set different proportions of users in the artificial society. The experimental results reveal that unfairness becomes more pronounced when there are more male users. We believe this is because the dataset contains more male user ratings. To validate this hypothesis, we use different proportions of data for training. As the ratio of male to female rating data increases from 1:1 to 2:1, and then to 3:1, the corresponding fairness performance also increases from 0.3602 to 0.3641, and finally to 0.3917.

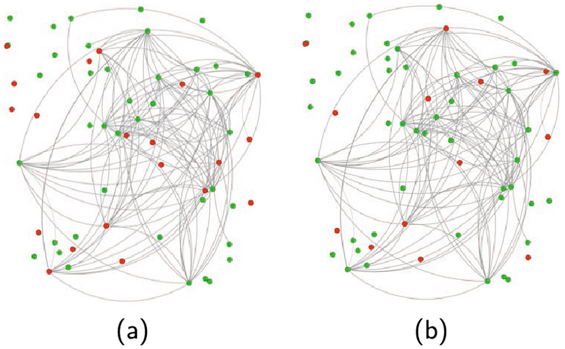


Fig. 5. Fairness comparison of recommendations accepted by users. (a) With the unfair recommendation algorithm. (b) With the fair recommendation algorithm.

5 Conclusion

This paper proposes EMAB, a method for evaluating the ethical behavior of algorithms based on computational experiments. A simulated test environment is constructed, which is dynamic, feedback-driven, and evolving. The experimental results demonstrate that the fairness performance of the algorithm deteriorates gradually in this environment. This highlights the limitations of static data and the need for EMAB. Our method is suitable for algorithms that operate in dynamic environments and interact with people. The current method is relatively simple for user behavior modeling. Future more complex user behavior will be considered. This method provides a new approach to the ethical evaluation of algorithmic behavior, where there is little similar work, effectively addressing the challenge of practical evaluation in complex situations.

Acknowledgements. This work has been supported in part by the National Key Research and Development Program of China (No.2021YFF0900800), National Natural Science Foundation of China (No.61972276, No.62206116, No.61832014, No.62032016), New Liberal Arts Reform and Practice Project of National Ministry of Education (No.2021170002), Tianjin University Talent Innovation Reward Program for Literature and Science Graduate Student(C1-2022-010).

References

1. Ji, L.: How to crack the information cocoon room under the background of intelligent media. Int. J. Soc. Sci. Educ. Res. **3**(3), 169–173 (2020)
2. Durán, J.M., Jongsma, K.R.: Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical AI. J. Med. Ethics **47**(5), 329–335 (2021)
3. Köchling, A., Wehner, M.C.: Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development. Bus. Res. **13**(3), 795–848 (2020)
4. Floridi, L.: Establishing the rules for building trustworthy AI. Nat. Mach. Intell. **1**(6), 261–262 (2019)
5. Bellamy, R.K., et al.: Ai fairness 360: an extensible toolkit for detecting and mitigating algorithmic bias. IBM J. Res. Dev. **63**(4/5), 4–1 (2019)
6. Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., Wilson, J.: The what-if tool: interactive probing of machine learning models. IEEE Trans. Visual Comput. Graphics **26**(1), 56–65 (2019)
7. D'Amour, A., Srinivasan, H., Atwood, J., Baljekar, P., Sculley, D., Halpern, Y.: Fairness is not static: deeper understanding of long term fairness via simulation studies. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 525–534 (2020)
8. Xue, X., et al.: Computational experiments: past, present and perspective. Acta Automatica Sinica **49**(2), 246–271 (2023)
9. Peng, H., Liu, C.: Breaking the information cocoon: when do people actively seek conflicting information? Proc. Assoc. Inf. Sci. Technol. **58**(1), 801–803 (2021)
10. Liu, W., Long, S., Xie, D., Liang, Y., Wang, J.: How to govern the big data discriminatory pricing behavior in the platform service supply chain? an examination with a three-party evolutionary game model. Int. J. Prod. Econ. **231**, 107910 (2021)
11. Berman, R.: Predictive algorithms in the criminal justice system: evaluating the racial bias objection. J. Phil. Polit. Econ. **126** (2017)
12. Stratigi, M., Nummenmaa, J., Pitoura, E., Stefanidis, K.: Fair sequential group recommendations. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing, pp. 1443–1452 (2020)
13. Raji, I.D., et al.: Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 33–44 (2020)
14. Raji, I.D., Buolamwini, J.: Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial AI products. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 429–435 (2019)
15. Bird, S., et al.: Fairlearn: a toolkit for assessing and improving fairness in AI. Microsoft, Technical Report. MSR-TR-2020-32 (2020)
16. Darias, J.M., Díaz-Agudo, B., Recio-Garcia, J.A.: A systematic review on model-agnostic xai libraries. In: ICCBR Workshops, pp. 28–39 (2021)
17. Zhou, D., Xue, X., Zhou, Z.: Sle2: the improved social learning evolution model of cloud manufacturing service ecosystem. IEEE Trans. Ind. Inf. **18**(12), 9017–9026 (2022)
18. Li, L., Huang, W.-L., Liu, Y., Zheng, N.-N., Wang, F.-Y.: Intelligence testing for autonomous vehicles: a new approach. IEEE Trans. Intell. Veh. **1**(2), 158–166 (2016)

19. Xue, X., Wang, S., Zhang, L., Feng, Z., Guo, Y.: Social learning evolution (sle): computational experiment-based modeling framework of social manufacturing. IEEE Trans. Ind. Inf. **15**(6), 3343–3355 (2018)
20. Xue, X., et al.: Computational experiments for complex social systems-part iii: the docking of domain models. IEEE Trans. Comput. Soc. Syst. (2023)
21. Lu, M., et al.: Computational experiments for complex social systems-part ii: the evaluation of computational models. IEEE Trans. Comput. Soc. Syst. **9**(4), 1224–1236 (2021)
22. Xue, X., Chen, F., Zhou, D., Wang, X., Lu, M., Wang, F.-Y.: Computational experiments for complex social systems-part i: the customization of computational model. IEEE Trans. Comput. Soc. Syst. **9**(5), 1330–1344 (2021)
23. Xue, X., et al.: Research roadmap of service ecosystems: a crowd intelligence perspective. Int. J. Crowd Sci. **6**(4), 195–222 (2022)
24. Ge, Y., Song, Z., Meng, R.: The method summary of generating large-scale artificial population in an artificial society. J. Syst. Simul. **31**(10), 1951 (2019)
25. Chen, B., et al.: Prediction of epidemic transmission and evaluation of prevention and control measures based on artificial society. J. Syst. Simul. **32**(12), 2507 (2020)
26. Bindra, S., Sharma, D., Parameswar, N., Dhir, S., Paul, J.: Bandwagon effect revisited: a systematic review to develop future research agenda. J. Bus. Res. **143**, 305–317 (2022)
27. Templeton, A., Neville, F.: Modeling collective behaviour: insights and applications from crowd psychology. In: Crowd Dynamics: Theory, Models, and Applications, vol. 2, pp. 55–81 (2020)
28. Chen, S., Liu, Y., Li, L.: Social selection-aware social network generation model. J. Syst. Eng. **34**(5), 587–597 (2019). (in Chinese)
29. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TIIS) **5**(4), 1–19 (2015)
30. Welling, M., Kipf, T.N.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR 2017) (2016)
31. Sedgwick, P.: Pearson's correlation coefficient. Bmj **345** (2012)

Metrics for Evaluating Actionability in Explainable AI

Hissah Alotaibi[1](✉) and Ronal Singh[2,3]

[1] University of Melbourne, Jizan, Kingdom of Saudi Arabia
hissahalotaibi5@gmail.com
[2] University of Melbourne, Melbourne, Australia
singhrr@unimelb.edu.au
[3] CSIRO's Data61, Alexandria, Australia
ronal.singh@csiro.au

Abstract. To enable recourse, explanations provided to people should be actionable, that is, explain what a person should do to change the model's decision. However, what actionability means in the context of explainable AI is unclear. In this paper, we explore existing tools that others developed to evaluate actionability in their respective domains. To our knowledge, no prior work in the XAI field has developed such a tool to evaluate the actionability of explanation. We conducted an experimental study to validate two existing actionability tools for discriminating the actionability of two types of explanations. Our results indicate that the two existing actionability tools reveal metrics relevant for conceptualising actionability for the XAI community.

Keywords: Algorithmic decision-making · explanation · actionability · metrics · machine learning

1 Introduction

Explainability is vital in domains like finance, medical diagnoses, and judicial decisions [2–4]. Given the rapid adoption of decision-making systems, models must offer both accuracy and actionable explanations for recourse, that is, to help users understand what they could do or change in the future to get a desirable outcome [8,9,11].

Previous studies suggest that counterfactual explanations could aid recourse, possibly requiring multiple of these to accommodate diverse backgrounds [6,9–11]. While multiple counterfactuals offer insights into favourable outcomes, they do not detail the actions for achieving them. Consequently, others propose explanations with action recommendations, guiding users on reaching the counterfactual state [2,8].

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-99-7022-3_44.

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 481–487, 2024.
https://doi.org/10.1007/978-981-99-7022-3_44

These suggestions from prior works to enable recourse often rely on researchers' understanding of what constitutes an 'actionable' explanation. However, no metrics for evaluating explanation actionability in XAI exist. We aim to identify such metrics by leveraging validated tools from other domains. We selected two validated actionability tools for patient education materials (PEMAT) [7] and cybersecurity advice [5]. In an experimental study, we assessed whether these metrics effectively gauge actionability in the directive [8] and counterfactual explanations [11]. We chose these two explanations because we believed that directive explanations were more actionable as they included explicit recommendations on what a user could do to change the outcome.

We ran an online study on Amazon MTurk with 90 participants in which we exposed participants to lending and employee turnover scenarios. Our results show that some of the metrics from existing actionability tools effectively assessed the two explanations' actionability. This suggests that the metrics these tools cover are relevant to how we conceptualise actionability for XAI. The ratings also suggested that counterfactual and directive explanations are *almost* equally actionable; they differed only on one metric across the two domains. Therefore, we could differentiate between the two explanations using the metrics from existing actionability tools. We find that having information that allows users to identify steps they would take is an important criterion and that making this information explicit is perhaps important. We also learn that some metrics may be domain or individual-dependent.

2 Study Design and Methodology

To provide an actionable explanation, we aim to identify metrics helpful in developing an actionability assessment tool for XAI. Our independent variable was the explanation type. Our dependent variable was the rating we collected using the metrics inspired by the PEMAT and cybersecurity tools. We manipulated our independent variables under the same control factors in three conditions to see the effect of presenting explanations on rating actionability metrics. Experimental conditions one and two were used to test differences between responses under different explanation conditions. We assumed that direct comparisons between explanation types might affect actionability perceptions. Therefore, we added experimental condition three to test whether exposure to multiple explanation types for a single scenario would have different effects than repeated exposure to a single explanation type.

- Condition C1: All three explanation options are counterfactual explanations (CF) [11]. For example, a counterfactual explanation could explain the minimum income needed to approve a loan. One of the options in each condition was an attention check.
- Condition C2: All three explanation options are directive explanation (DX) [8]. For example, to increase their income, the customer might be recommended to rent a room in their house.

- Condition C3: Explanation options are of different types - one directive explanation and the other counterfactual explanation.

2.1 Scenarios

We designed eight scenarios: four for credit risk assessment and four for employee turnover decisions (refer to Appendices A and B for scenario details). Using the Lending Club dataset[1], we trained a logistic regression model for loan default prediction, and another model for employee resignation prediction using the employee turnover dataset[2]. These models achieved 85% and 80% accuracy on the respective datasets. After training, we selected four customer/employee records for scenario creation. We generated counterfactual and directive explanations by employing existing techniques: [6] for the counterfactual and [8] for the directive explanations.

Each scenario consisted of three parts: a customer/employee profile with relevant details, an incomplete explanation including the AI decision (e.g., loan approval/denial), and participant instructions. The participants selected one of the three explanations they thought they could use to change an AI decision in the future (e.g. from denying to approving a loan). In the credit domain, participants acted as customers interacting with the AI. In contrast, in the employee turnover domain, they played the role of an employee's supervisor engaging with the AI system. All scenarios concluded with an adverse outcome for the individual.

2.2 Actionability Metrics

We used actionability metrics from PEMAT and the cybersecurity questionnaire. We did not include audio and visual metrics (items) from PEMAT because we provided primarily textual explanations. We asked participants to rate the following three questions (answered on a scale of 0 (completely disagree) to 10 (completely agree):

IT1 My chosen explanation **clearly** identifies **at least one action** I can take.

IT2 My chosen explanation **addresses me directly** when describing what to do.

IT3 My chosen explanation breaks down any action into **manageable, explicit steps**.

We modified and incorporated statements from the cybersecurity questionnaire into the scenarios. Participants assessed these using the same scale (see above).

IT4 I considered how **difficult** it would be for me to implement the chosen explanation.

[1] https://www.kaggle.com/datasets/husainsb/lendingclub-issued-loans.

[2] https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset.

IT5 I considered the amount of **time** it would take me to implement the chosen explanation.

IT6 I considered the amount of **disruption** it would cause me to implement the chosen explanation.

IT7 I considered the level of **confidence** I have in implementing the chosen explanation.

2.3 Online Study

We conducted an online study with 90 individuals (30 participants in each experimental condition) using **Amazon MTurk**. The experiment was designed as a Qualtrics[3] survey. Ethics approval was obtained from our university before the experiment. Participants were compensated USD $7 for participating in the experiment.

The survey had five steps, adapted from [1,8]. The participants were first given a plain language statement and then a consent form. If participants agreed to continue, they answered logical questions to remove the robotic respondents and filled out a demographic questionnaire. Following this, participants were randomly assigned to one of three explanation conditions. These three conditions had the same scenarios and questions, but the difference was just the type of explanation. Next, participants were provided with instructions and then scenarios. The main tasks involved four scenarios; two of the four were randomly selected from credit and two from the employee domain. The order of scenarios and domains was randomised to reduce ordering effects.

We asked participants two questions in each scenario. The first question asked the participants to select one explanation (of the three, with one being the attention check) they thought was most actionable for them to use to reach a desirable outcome (e.g. approved loan) in the future. The explanation options were randomised to reduce ordering effects. The second question asked the participants to rate their chosen explanation using the actionability questions discussed earlier. Data from the Qualtrics survey was first transferred to R software for analysis. The Mann-Whitney U test was used to identify the differences in the rating between metrics and Cohen's d for effect size measure.

3 Results and Discussion

Our results show that some of the metrics from existing actionability tools effectively assessed the two explanations' actionability. The ratings also suggested that counterfactual and directive explanations are *almost* equally actionable; they differed only on one metric across the two domains. Therefore, we could differentiate between the two explanations using the metrics from existing actionability tools. We used the attention check question to remove the three participants who were not engaged with the experiment. Of the 90, we had 87 participants after removal, with 27 in condition 1, 26 in condition 2 and 34 in condition 3. The average study time was 29 mins (SD = 11).

[3] https://www.qualtrics.com/au/.

All participants were from the United States. 51.7% were Male ,46.0% were female and 1.1% did not state their gender. In terms of age, 19.5% were 25–34, 33.3% were 35–44, 29.9% were 45–54, 13.8% were 55–74, and the rest were above 65 (3.4%). Regarding education, 13.8% were High school graduates, 18.8% had some college but no degree, 48.3% had an Associate or Bachelor's degree, 5.7% had a Master's degree, and 1.1% had a Professional degree. Participants experienced applying for a credit comprised Not familiar at all (4.6%), Slightly familiar(17.2%), Moderately familiar (32.2%), very familiar (31.0%) and Extremely familiar (14.9%). For familiarity with human resource management, 13.8% were Not familiar at all, 35.6% were Slightly Familiar, 31.0% were Moderately familiar, 13.8% were very familiar, and 5.7% were extremely familiar.

3.1 Experimental Condition One and Two

We show the ratings for the two explanation types in both domains in Fig. 1 (we report detailed statistics in the Appendix). We observed that the median of each actionability metric was slightly higher in the directive explanation (DX) than in the counterfactual explanation (CF). Hence, it is evident that DX had a positive trend across most actionability metrics.

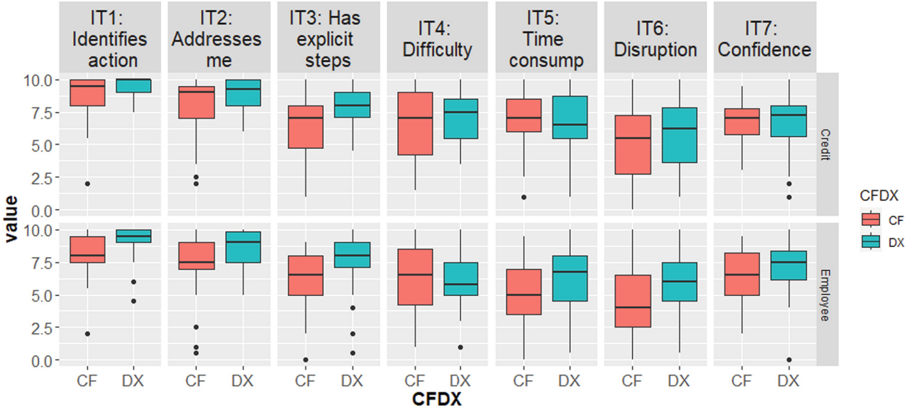


Fig. 1. Comparing actionability metrics; credit domain show on top.

The Mann-Whitney U test indicated that, in both domains, the directive explanation (DX) received significantly higher ratings than the counterfactual explanation (CF) for the metric "breaks down any action into manageable, explicit steps" (credit: $W = 493.5, p = 0.01$; employee: $W = 506, p = 0.005$). This highlights DX's provision of more detailed information, aiding users in identifying actionable steps more effectively than the CF model. Furthermore, domain distinctions influenced explanation assessments. In the employee turnover domain, the Mann-Whitney U test showed that DX had significantly higher ratings than

CF for two additional metrics inspired by PEMAT: "clearly identifies at least one action" ($W = 492, p = 0.009$) and "explanation addresses me directly" ($W = 457, p = 0.05$). This suggests that counterfactual explanations include implicit actions, while directive explanations explicitly state them.

However, the cybersecurity tool did not differentiate between the two explanation types. One reason could be that the metrics from the cybersecurity questionnaire were more relevant to the specific action someone would take. At the same time, the PEMAT had metrics that judged whether the explanation had information for someone to identify the action in the first place. This suggests that the three PEMAT-inspired metrics are essential in conceptualising and developing an actionability assessment tool for XAI (Table 1).

Table 1. Comparing the actionability metrics inspired by PEMAT and cybersecurity between C1 and C2 by Mann Whitney U Test and Cohen's d calculations in credit and employee domains.

Actionability metrics	Credit			Employee turnover		
	W	p-value	Cohen's d	W	p-value	Cohen's d
Clearly identifies at least one action	449	0.06	–	492	**0.0096**	0.67 (medium)
Addresses me directly	422	0.20	–	457	**0.05**	0.60 (medium)
Breaks down any action	493.5	**.01**	–0.80 (large)	506	**0.005**	0.68 (medium)
Difficulty	376	0.66	–	319	0.57	–
Time Consumption	307.5	0.44	–	430.5	0.15	–
Disruption	409.5	0.30	–	449.5	0.07	–
Confidence	381	0.59	–	411.5	0.28	–

3.2 Experimental Condition Three

In condition three, most participants favoured directives over counterfactual explanations. Of 68 responses, 58 in the credit domain and 63 in the employee domain chose DX. Consequently, a direct comparison between DX and CF in C3 is unfeasible due to the limited number of participants who selected CF. Nevertheless, we compared C2 (all DX options) with C3 by excluding a few CF respondents. A Mann-Whitney U test demonstrated a significant disparity in directive explanation ratings between conditions 2 and 3 in specific metrics, suggesting that participants' appreciation for directive explanations improved significantly when they were given counterfactual and directive explanations in the same scenario.

4 Conclusion

We evaluated two existing actionability tools to test whether these metrics effectively assess the actionability of the two types of explanations, directive and counterfactual. We identified three items that effectively assessed the actionability of directive and counterfactual explanations. We hope these metrics assist designers in creating more actionable explanations that allow end users to act effectively. We recommend that future works consider our results as a starting point to develop an actionability assessment tool for XAI.

References

1. Binns, R., Van Kleek, M., Veale, M., Lyngs, U., Zhao, J., Shadbolt, N.: 'it's reducing a human being to a percentage' perceptions of justice in algorithmic decisions. In: Proceedings of the 2018 Chi Conference on Human Factors in Computing Systems, pp. 1–14 (2018)
2. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2021, pp. 353–362. Association for Computing Machinery, New York (2021)
3. Kononenko, I.: Machine learning for medical diagnosis: history, state of the art and perspective. Artif. Intell. Med. **23**(1), 89–109 (2001)
4. Peng, A., Simard-Halm, M.: The perils of objectivity: towards a normative framework for fair judicial decision-making. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 343–343 (2020)
5. Redmiles, E.M., et al.: A comprehensive quality evaluation of security and privacy advice on the web. In: 29th USENIX Security Symposium (USENIX Security 2020), pp. 89–108 (2020)
6. Russell, C.: Efficient search for diverse coherent explanations. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 20–28 (2019)
7. Shoemaker, S.J., Wolf, M.S., Brach, C.: Development of the patient education materials assessment tool (pemat): a new measure of understandability and actionability for print and audiovisual patient information. Patient Educ. Couns. **96**(3), 395–403 (2014)
8. Singh, R., et al.: Directive explanations for actionable explainability in machine learning applications. ACM Trans. Interact. Intell. Syst. (2023). https://doi.org/10.1145/3579363
9. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 10–19 (2019)
10. Venkatasubramanian, S., Alfano, M.: The philosophical basis of algorithmic recourse. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* 2020, pp. 284–293. Association for Computing Machinery, New York (2020)
11. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. JL Tech. **31**, 841 (2017)

Author Index

F. Liu et al. (Eds.): PRICAI 2023, LNAI 14326, pp. 489–494, 2024.
https://doi.org/10.1007/978-981-99-7022-3