

Chengqing Zong
Jian-Yun Nie
Dongyan Zhao
Yansong Feng (Eds.)

Communications in Computer and Information Science

496

Natural Language Processing and Chinese Computing

Third CCF Conference, NLPCC 2014
Shenzhen, China, December 5–9, 2014
Proceedings

 Springer



Editorial Board

Simone Diniz Junqueira Barbosa

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rio de Janeiro, Brazil*

Phoebe Chen

La Trobe University, Melbourne, Australia

Alfredo Cuzzocrea

ICAR-CNR and University of Calabria, Italy

Xiaoyong Du

Renmin University of China, Beijing, China

Joaquim Filipe

Polytechnic Institute of Setúbal, Portugal

Orhun Kara

TÜBİTAK BİLGEM and Middle East Technical University, Turkey

Igor Kotenko

*St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences, Russia*

Krishna M. Sivalingam

Indian Institute of Technology Madras, India

Dominik Ślęzak

University of Warsaw and Infobright, Poland

Takashi Washio

Osaka University, Japan

Xiaokang Yang

Shanghai Jiao Tong University, China

Chengqing Zong Jian-Yun Nie
Dongyan Zhao Yansong Feng (Eds.)

Natural Language Processing and Chinese Computing

Third CCF Conference, NLPCC 2014
Shenzhen, China, December 5-9, 2014
Proceedings



Springer

Volume Editors

Chengqing Zong
Chinese Academy of Sciences, Beijing, China
E-mail: cqzong@nlpr.ia.ac.cn

Jian-Yun Nie
University of Montreal, Montreal, QC, Canada
E-mail: nie@iro.umontreal.ca

Dongyan Zhao
Peking University, Beijing, China
E-mail: zhaodongyan@pku.edu.cn

Yansong Feng
Peking University, Beijing, China
E-mail: fengyansong@pku.edu.cn

ISSN 1865-0929

e-ISSN 1865-0937

ISBN 978-3-662-45923-2

e-ISBN 978-3-662-45924-9

DOI 10.1007/978-3-662-45924-9

Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014957373

© Springer-Verlag Berlin Heidelberg 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are very happy to welcome you to the proceedings of NLPCC 2014, the Third International Conference on Natural Language Processing and Chinese Computing. NLPCC is the annual conference of CCF-TCCI (Technical Committee of Chinese Information, China Computer Federation). The first and the second NLPCC conferences were successfully held in Beijing in 2012 and Chongqing in 2013. As a leading conference in the field of NLP and Chinese computing, NLPCC has been the premier forum for Chinese researchers and practitioners in NLP from academia, industry, and government to share their ideas, research results, and experiences, and to promote the research and technical innovations in these fields. While the first two NLPCC conferences focused on research in China, this year, NLPCC became an international conference. Contributions from not only China, but also Pacific Asia areas and all over the world are solicited.

We received 226 paper submissions this year, including 110 manuscripts written in English and 116 written in Chinese. Among them, 48 (21.23%) were accepted as long papers (35 in English and 13 in Chinese) and 19 (8.40%) as posters (eight in English and 11 in Chinese). All submissions were double-blind reviewed by three reviewers. The final decisions were made at a meeting with Program Committee (PC) chairs and area chairs. Among the accepted papers, seven were recommended by area chairs as candidates for best papers. The final selection of the best papers was made by an independent best paper committee. These proceedings only include the accepted English papers; the accepted Chinese papers are published by ACTA Scientiarum Naturalium Universitatis Pekinensis.

NLPCC 2014 featured three distinguished invited speakers: Eduard Hovy (Carnegie Mellon University), Bing Liu (University of Illinois at Chicago), and Hwee Tou Ng (National University of Singapore).

The success of NLPCC is due to the great contributions of a large number of people. We would like to thank all the area chairs for their hard work for recruiting reviewers, monitoring the review and discussion processes, and carefully rating and recommending submissions. We would like to thank all 154 reviewers for their time and efforts in reviewing the submissions. We are very grateful to Dekang Lin and Ming Zhou for their participation in the best paper committee. We are also grateful for the help and support of the general chairs Dekang Lin and Guodong Zhou and the Organization chairs Dongyan Zhao and Qincui Chen. Special thanks go to Yansong Feng, the publication chair, and Jiajun Zhang, the demo chair, for their great help. We appreciate it very much!

We would like to thank all the authors who submitted their research to the conference.

Finally, we would also like to thank our sponsors for their contributions to the conference.

We look forward to seeing you at NLPCC 2014 in Shenzhen and hope you enjoy the conference!

December 2014

Chengqing Zong
Jian-Yun Nie

Organization

Organizing Committee

General Chairs

Dekang Lin

Google

Guodong Zhou

Soochow University, China

Program Committee Co-chairs

Chengqing Zong

Institute of Automation, CAS, China

Jian-Yun Nie

Montreal University, Canada

Area Co-chairs

Fundamentals on CIT

Yue Zhang

Singapore University of Technology and Design

Fang Kong

Soochow University, China

Applications on CIT

Ying Zhang

Facebook/Carnegie Mellon University, USA

Shuanhu Bai

Sina

Machine Translation

Yang Liu

Tsinghua University, China

Fei Huang

IBM T.J. Watson Research Center

Web Mining and Big Data

Xiaojun Wan

Peking University, China

Wenjie Li

The Hong Kong Polytechnic University

Machine Learning for NLP

Jun Zhu

Tsinghua University, China

Xiaojin Zhu

University of Wisconsin-Madison, USA

Knowledge Acquisition

Zhifang Sui

Peking University, China

Shumin Shi

Microsoft Research Asia

NLP for Social Networks

Xiaohua Liu
Huawei Shen

University of Montreal, Canada
ICT, CAS, China

NLP for Search and Ads

Hongfei Lin
Fangtao Li

Dalian Institute of Technology, China
Google

QA and Information Extraction

Heng Ji
Qi Zhang

Rensselaer Polytechnic Institute, USA
Fudan University, China

Panel Chair

Ming Zhou

Microsoft Research Asia

Demo Co-chairs

Kuo Zhang
Jiajun Zhang

Sogou
Institute of Automation, CAS, China

Organization Co-chairs

Dongyan Zhao
Qingcai Chen

Peking University, China
Shenzhen Graduate School of HIT, China

Best Paper Awards Committee

Jianyun Nie
Ming Zhou
Dekang Lin

Montreal University
Microsoft Research Asia
Google

Sponsor Co-chairs

Zhi Tang
Xuan Wang

Peking University, China
Shenzhen Graduate School of HIT, China

Publicity Co-chairs

Yangsen Zhang
Fei Xia
Feiyu Xu

Beijing Information Science and Technology
University, China
The University of Washington, USA
DFKI, Germany

ADL/Tutorial Co-chairs

Jun Zhao	Institute of Automation, CAS, China
Min Zhang	Tsinghua University, China

Publication Chair

Yansong Feng	Peking University, China
--------------	--------------------------

Website Chair

Aixia Jia	Peking University, China
-----------	--------------------------

Program Committee

Bai, Shuanhu	Sina
Chen, Boxing	National Research Council Canada
Chen, Jiajun	Nanjing University, China
Chen, Lin	Intellius
Chen, Lin	Xiamen University, China
Chen, Wenliang	Soochow University, China
Chen, Yidong	Xiamen University, China
Chen, Yueguo	Renmin University of China
Chen, Yufeng	Institute of Automation, CAS, China
Dai, Xinyu	Nanjing University, China
Deng, Zhihong	Peking University, China
Dou, Zhicheng	Microsoft Research Asia
Duan, Nan	Microsoft Research Asia
Duan, Xiangyu	Soochow University, China
Feng, Minwei	IBM
Feng, Yansong	Peking University, China
Han, Xianpei	Institute of Software, CAS, China
He, Zhongjun	Baidu
Hong, Yu	Soochow University, China
Hu, Yunhua	Alibaba Corp.
Huang, Fei	Carnegie-Mellon University, USA
Huang, Hongzhao	RPI
Huang, Shujian	Nanjing University, China
Ji, Donghong	Wuhan University, China
Jia, Yuxiang	Zhengzhou University, China
Jiang, Hongfei	Alibaba Corp.
Jiang, Jing	Singapore Management University
Jiang, Minghu	Tsinghua University, China
Jiang, Wenbin	Institute of Computing Technology, CAS, China
Jin, Peng	Leshan Normal University, China
Lam, Wai	The Chinese University of Hong Kong

Li, Baoli	Henan University of Technology, China
Li, Haibo	Nuance
Li, Hao	RPI
Li, Maoxi	Jiangxi Normal University, China
Li, Peifeng	Soochow University, China
Li, Qingsheng	Anyang Normal University, China
Li, Sujian	Peking University, China
Li, Wenjie	Hong Kong Polytechnic University
Li, Wujun	Nanjing University, China
Liu, Chenglin	Institute of Automation, CAS, China
Liu, Kang	Institute of Automation, CAS, China
Liu, Maofu	Wuhan University of Science and Technology, China
Liu, Pengyuan	Beijing Language and Culture University, China
Liu, Qun	Dublin City University, Ireland
Liu, Yang	Tsinghua University, China
Liu, Yang	Shandong University, China
Liu, Yiqun	Tsinghua University, China
Liu, Zhiyuan	Tsinghua University, China
Lv, Yajuan	Baidu
Matthias Eck	Facebook
Qi, Haoliang	Heilongjiang Institute of Technology, China
Qin, Bing	Harbin Institute of Technology, China
Qiu, Likun	Ludong University, China
Qu, Weiguang	Nanjing Normal University, China
Ren, Feiliang	Northeastern University
Shao, Yanqiu	Beijing Language and Culture University, China
Shen, Dou	Baidu
Shen, Huawei	Institute of Computing, CAS, China
Shen, Libin	Mobvoi Inc.
Shi, Xiaodong	Xiamen University, China
Sun, Aixin	Nanyang Technological University, Singapore
Tang, Zhi	Peking University, China
Wan, Xiaojun	Peking University, China
Wang, Zhichun	Beijing Normal University, China
Wang, Zhiguo	Brandeis University, USA
Wu, Hua	Baidu
Xiang, Bing	Reuters
Xiao, Xinyan	Microsoft STCA
Xiong, Deyi	Soochow University, China
Xu, Jinan	Beijing Jiaotong University, China

Xu, Jun	Noah's Ark Lab of Huawei Technologies
Xu, Ruifeng	Harbin Institute of Technology Shenzhen Graduate School, China
Xu, Weiran	PRIS
Yan, Yulan	NICT, Japan
Yang, Jianwu	Peking University, China
Yang, Muyun	Harbin Institute of Technology, China
Yu, Zhengtao	Kunming University of Science and Technology, China
Zan, Hongying	Zhengzhou University, China
Zhang, Dakun	Toshiba China
Zhang, Dongdong	Microsoft Research Asia
Zhang, Jiajun	Institute of Automation, CAS, China
Zhang, Min	Tsinghua University, China
Zhang, Min	Soochow University, China
Zhao, Dongyan	Peking University, China
Zhao, Hai	Shanghai Jiao Tong University, China
Zhao, Shiqi	Baidu
Zhao, Tiejun	Harbin Institute of Technology, China
Zhou, Ming	Microsoft Research Asia
Zhou, Qiang	Tsinghua University, China
Zhou, Yu	Institute of Automation, CAS
Zhu, Jingbo	Northeastern University
Zhu, Xiaodan	National Research Council Canada

Organizers

Organized by



China Computer Federation, China

Hosted by



Shenzhen Graduate School of HIT

数字出版技术
国家重点实验室

State Key Laboratory of Digital Publishing

In Cooperation with:



ACTA Scientiarum Naturalium
Universitatis Pekinensis



Springer

the language of science

Springer

Sponsoring Institutions

Microsoft®

Research

微软亚洲研究院

Microsoft Research Asia

Baidu 百度

Baidu Inc.

Sogou 搜狗

Sogou Inc.

新浪微博
weibo.com

Sina Weibo

Table of Contents

Long Papers

Fundamentals on Language Computing

A Global Generative Model for Chinese Semantic Role Labeling	1
<i>Haitong Yang and Chengqing Zong</i>	
Chinese Comma Disambiguation on K-best Parse Trees	13
<i>Fang Kong and Guodong Zhou</i>	
Event Schema Induction Based on Relational Co-occurrence over Multiple Documents	23
<i>Tingsong Jiang, Lei Sha, and Zhifang Sui</i>	
Negation and Speculation Target Identification	34
<i>Bowei Zou, Guodong Zhou, and Qiaoming Zhu</i>	

Applications on Language Computing

An Adjective-Based Embodied Knowledge Net	46
<i>Chang Su, Jia Tian, and Yijiang Chen</i>	
A Method of Density Analysis for Chinese Characters	54
<i>Jingwei Qu, Xiaoqing Lu, Lu Liu, Zhi Tang, and Yongtao Wang</i>	
Computing Semantic Relatedness Using a Word-Text Mutual Guidance Model	67
<i>Bingquan Liu, Jian Feng, Ming Liu, Feng Liu, Xiaolong Wang, and Peng Li</i>	
Short Text Feature Enrichment Using Link Analysis on Topic-Keyword Graph	79
<i>Peng Wang, Heng Zhang, Bo Xu, Chenglin Liu, and Hongwei Hao</i>	

Machine Translation and Multi-Lingual Information Access

Sentence-Length Informed Method for Active Learning Based Resource-Poor Statistical Machine Translation	91
<i>Jinhua Du, Miaomiao Wang, and Meng Zhang</i>	

Detection of Loan Words in Uyghur Texts	103
<i>Chenggang Mi, Yating Yang, Lei Wang, Xiao Li,</i> <i>and Kamali Dalielihan</i>	
A Novel Rule Refinement Method for SMT through Simulated Post-Editing	113
<i>Sitong Yang, Heng Yu, and Qun Liu</i>	
Case Frame Constraints for Hierarchical Phrase-Based Translation: Japanese-Chinese as an Example	123
<i>Jiangming Liu, JinAn Xu, Jun Xie, and Yujie Zhang</i>	

Machine Learning for NLP

Bridging the Language Gap: Learning Distributed Semantics for Cross-Lingual Sentiment Classification	138
<i>Guangyou Zhou, Tingting He, and Jun Zhao</i>	
A Short Texts Matching Method Using Shallow Features and Deep Features	150
<i>Longbiao Kang, Baotian Hu, Xiangping Wu, Qingcai Chen,</i> <i>and Yan He</i>	
A Feature Extraction Method Based on Word Embedding for Word Similarity Computing	160
<i>Weitai Zhang, Weiran Xu, Guang Chen, and Jun Guo</i>	
Word Vector Modeling for Sentiment Analysis of Product Reviews	168
<i>Yuan Wang, Zhaohui Li, Jie Liu, Zhicheng He, Yalou Huang,</i> <i>and Dong Li</i>	
Cross-Lingual Sentiment Classification Based on Denoising Autoencoder	181
<i>Huiwei Zhou, Long Chen, and Degen Huang</i>	

NLP for Social Media

Aspect-Object Alignment Using Integer Linear Programming	193
<i>Yanyan Zhao, Bing Qin, and Ting Liu</i>	
Sentiment Classification of Chinese Contrast Sentences	205
<i>Junjie Li, Yu Zhou, Chunyang Liu, and Lin Pang</i>	
Emotion Classification of Chinese Microblog Text via Fusion of BoW and eVector Feature Representations	217
<i>Chengxin Li, Huimin Wu, and Qin Jin</i>	

Social Media as Sensor in Real World: Geolocate User with Microblog	229
<i>Xueqin Sui, Zhumin Chen, Kai Wu, Pengjie Ren, Jun Ma, and Fengyu Zhou</i>	

A Novel Calibrated Label Ranking Based Method for Multiple Emotions Detection in Chinese Microblogs	238
<i>Mingqiang Wang, Mengting Liu, Shi Feng, Daling Wang, and Yifei Zhang</i>	

Enhance Social Context Understanding with Semantic Chunks.....	251
<i>Siqiang Wen, Zhixing Li, and Juanzi Li</i>	

NLP for Search Technology and Ads

Estimating Credibility of User Clicks with Mouse Movement and Eye-Tracking Information	263
<i>Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma</i>	

<i>Cannabis-TREATS-Cancer</i> : Incorporating Fine-Grained Ontological Relations in Medical Document Ranking	275
<i>Yunqing Xia, Zhongda Xie, Qiuge Zhang, Huiyuan Wang, and Huan Zhao</i>	

A Unified Microblog User Similarity Model for Online Friend Recommendation	286
<i>Shi Feng, Le Zhang, Daling Wang, and Yifei Zhang</i>	

Weakly-Supervised Occupation Detection for Micro-blogging Users	299
<i>Ying Chen and Bei Pei</i>	

Normalization of Chinese Informal Medical Terms Based on Multi-field Indexing	311
<i>Yunqing Xia, Huan Zhao, Kaiyu Liu, and Hualing Zhu</i>	

Question Answering and User Interaction

Answer Extraction with Multiple Extraction Engines for Web-Based Question Answering	321
<i>Hong Sun, Furu Wei, and Ming Zhou</i>	

Answering Natural Language Questions via Phrasal Semantic Parsing	333
<i>Kun Xu, Sheng Zhang, Yansong Feng, and Dongyan Zhao</i>	

A Fast and Effective Method for Clustering Large-Scale Chinese Question Dataset 345
Xiaodong Zhang and Houfeng Wang

Web Mining and Information Extraction

A Hybrid Method for Chinese Entity Relation Extraction 357
Hao Wang, Zhenyu Qi, Hongwei Hao, and Bo Xu

Linking Entities in Tweets to Wikipedia Knowledge Base 368
Xianqi Zou, Chengjie Sun, Yaming Sun, Bingquan Liu, and Lei Lin

Automatic Recognition of Chinese Location Entity 379
Xuewei Li, Xueqiang Lv, and Kehui Liu

Detect Missing Attributes for Entities in Knowledge Bases via Hierarchical Clustering 392
Bingfeng Luo, Huanquan Lu, Yigang Diao, Yansong Feng, and Dongyan Zhao

Improved Automatic Keyword Extraction Based on TextRank Using Domain Knowledge 403
Guangyi Li and Houfeng Wang

Short Papers

A Topic-Based Reordering Model for Statistical Machine Translation ... 414
Xing Wang, Deyi Xiong, Min Zhang, Yu Hong, and Jianmin Yao

Online Chinese-Vietnamese Bilingual Topic Detection Based on RCRP Algorithm with Event Elements 422
Wen-xu Long, Ji-xun Gao, Zheng-tao Yu, Sheng-xiang Gao, and Xu-dong Hong

Random Walks for Opinion Summarization on Conversations 430
Zhongqing Wang, Liyuan Lin, Shoushan Li, and Guodong Zhou

TM-ToT: An Effective Model for Topic Mining from the Tibetan Messages 438
Chengxu Ye, Wushao Wen, and Ping Yang

Chinese Microblog Entity Linking System Combining Wikipedia and Search Engine Retrieval Results 449
Zeyu Meng, Dong Yu, and Endong Xun

Emotion Cause Detection with Linguistic Construction in Chinese Weibo Text	457
<i>Lin Gui, Li Yuan, Ruifeng Xu, Bin Liu, Qin Lu, and Yu Zhou</i>	
News Topic Evolution Tracking by Incorporating Temporal Information	465
<i>Jian Wang, Xianhui Liu, Junli Wang, and Weidong Zhao</i>	
Author Index	473

A Global Generative Model for Chinese Semantic Role Labeling

Haitong Yang and Chengqing Zong

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{htyang, cqzong}@nlpr.ia.ac.cn

Abstract. The predicate and its semantic roles compose a unified entity that conveys the semantics of a given sentence. A standard pipeline of current approaches to semantic role labeling (SRL) is that for a given predicate in a sentence, we can extract features for each candidate argument and then perform the role classification through a classifier. However, this process totally ignores the integrality of the predicate and its semantic roles. To address this problem, we present a global generative model in which a novel concept called Predicate-Arguments-Coalition (PAC) is proposed to encode the relations among individual arguments. Owing to PAC, our model can effectively mine the inherent properties of predicates and obtain a globally consistent solution for SRL. We conduct experiments on the standard benchmarks: Chinese PropBank. Experimental results on a single syntactic tree show that our model outperforms the state-of-the-art methods.

Keywords: global generative model, semantic role labeling, Predicate-Arguments-Coalition (PAC).

1 Introduction

Semantic Role Labeling (SRL) is a kind of shallow semantic parsing task and its goal is to recognize some related phrases and assign a joint structure (WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW) to each predicate of a sentence[5]. Because of its ability to encode semantic information, SRL has been applied in many tasks of NLP, such as question and answering[13], information extraction[3, 15] and machine translation[9, 21, 23, 28].

Since the release of FrameNet[1] and PropBank[7, 24], there has been a large amount of work on SRL[5, 10, 11, 18, 20, 22, 25, 26, 29]. When labeling the candidate arguments, a common pipeline schema works as: first extract features from a syntax tree and then independently accomplish the classification for each candidate. An implicit fact behind the process is that there is no interaction among the candidate arguments. However, from linguistic intuition this is not appropriate because an arguments frame of a predicate is a joint structure, with strong dependencies between arguments[19]. For example, if ARG0 is assigned to one argument, then the other arguments are not allowed to be classified into ARG0.

To address the argument dependencies in SRL, there has been some work[6, 19]. Their fundamental view is that the predicate argument structure is a sequence structure. In these approaches, they usually introduce label sequence features into the original model in order to capture the global properties of the arguments. They reported better performance than the original model, which does not consider the argument dependencies.

Different from the above viewpoints, we attempted to directly obtain a global structure for SRL. We introduce a novel concept called Predicate-Argument-Coalition (PAC) to describe the global structure of the predicate and arguments. PAC can naturally catch many linguistic phenomena about the predicates. Based on PAC, we propose a global generative model (GGM) that could obtain a globally consistent structure for SRL directly. Our model works according to the following schema:

- First, we train a base local classifier;
- Second, one binary classifier is trained to distinguish the core arguments from adjunction arguments;
- Third, traverse all of the possible PAC candidates, and then the candidate with the highest score is selected.

The PAC of the predicates could be considered to be one type of prior knowledge about the given predicate and it provides essential assistance for SRL. Our experimental results on Chinese PropBank show that GGM significantly outperforms the state-of-the-art systems and the relative error declines 13.8% compared with the baseline model. Furthermore, after a new feature Word Semantic Class is added, GGM achieves approximately one point of F1 score in improvement.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents our novel concept Predicate-Argument-Coalition. Section 4 presents a standard local classifier as the baseline. Our generative model GGM is described and formulated in Section 5. The experiments and results are presented in Section 6. The conclusions can be found in Section 7.

2 Related Work

Since the pioneering work of [5], there has been a large number of studies on automatic semantic role labeling. A variety of approaches[2, 11, 12, 16, 17, 27] have been successfully applied in SRL. Here we give a brief introduction to some typical studies. [25] takes a critical look at features of arguments detection and arguments classification. [17] discusses Chinese semantic role labeling with shallow parsing and [8] explores the joint syntactic and semantic parsing of Chinese to further improve the performance of both syntactic parsing and SRL.

There are some studies devoted to utilizing the global information of arguments and predicates to improve SRL. [14] constructed an Integer Linear Programming architecture in which the dependency relations among the arguments are implied by the constraints. They summarized ten categories of constraint conditions in which approximately one half of the categories are defined to describe the mutual dependences among the candidate arguments. However, one drawback of their system

is that some constraints must be provided by hand. For example, the 10th constraint of their system is that given the predicate, some argument classes are illegal. To provide the constraint, all predicates must be checked as to whether every argument class is legal for the predicates.

[19] proposes a joint model to explore the global information in the arguments. They first build a local model, then use the local model to generate the n most likely joint assignments for a sentence, and finally rerank these the n joint assignments through a joint model in which many global features are defined. In their model, almost all of the global features are expressed in the argument sequence's way such as using whole label sequence features, because in their view this type of sequence structure could handle the structure of the semantic roles. Their experiment results supported their viewpoint. However, we have different opinions about the structure of semantic roles. Unfortunately, according to our investigation, in some cases the change in the position of some particular arguments will not cause a change in the semantic meaning. For example, “明天 我 回去”(Tomorrow I will go back) and “我 明天 回去”(I will go back tomorrow) have the exact meaning and the same semantic roles but their label sequences are different. Therefore, it is debatable to use an ordered sequence structure to express the predicate-argument structure. In addition, the sequence structure is very sparse because there are over 20 types of labels in Prop-Bank. If the sequence length is n , the number of all possible sequences will be $20n$. In the model of [19], there are many lexicalization features that aggravate the sparsity.

It is also worth noting that the predicate is dominant in the predicate-argument structure and all arguments serve the predicate. However, in almost all of the existing approaches, the predicate is treated as only one feature of the discriminative model. In our opinion, every predicate has its own characteristics, and we should mine the intrinsic properties of the predicates to help SRL.

3 Predicate-Argument Structure

In the SRL community, the predicate-argument structure is thought to be a composite structure of the predicate and arguments. Figure 1 shows an example for the predicate-argument structure. In the sentence shown, the predicate is ‘提供’ (provide) and is labeled as “pred”. And there are five arguments for ‘提供’ (provide); three of these arguments are core arguments, which are labeled as ARG + a number, and the other two are adjunction arguments which are labeled as ARGM + an adjunction word. These two types of labels have different functions in conveying the meanings of the sentence. The core arguments are essential to constitute the skeleton of the sentence, while the adjunction arguments provide additional information for the predicate, such as Time, Location and so on.

However, there is still not an explicit formulation for representing the predicate-argument structure. In this paper, we propose a novel concept named Predicate-Argument-Coalition (PAC) to represent the predicate-argument structure. And we experimentally demonstrate that PAC is effective to represent the predicate-argument structure.

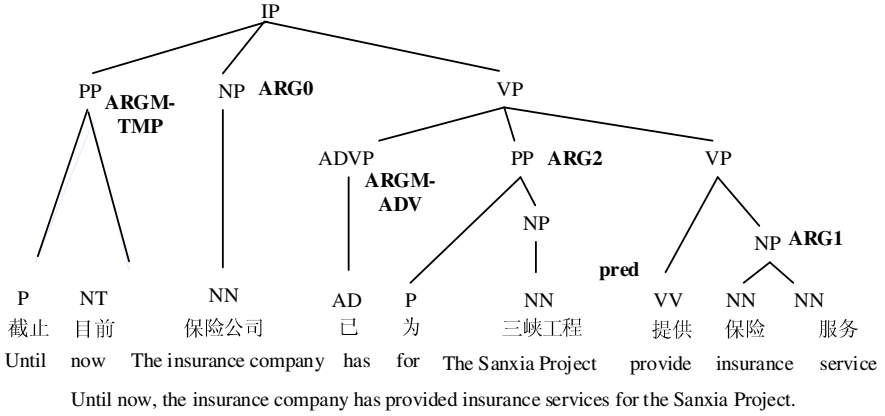


Fig. 1. An example from Chinese PropBank

3.1 Predicate-Argument-Coalition

Prediate-Argument-Coalition (PAC) is defined as a three-tuple like below.

$$PAC \triangleq \langle Pred, (ArgX : n), (ArgM : n) \rangle$$

The first term is the predicate; the second term is a multi-set that represents the entirety of the core arguments in which n is the number for $ArgX$, which appears in the sentence; the third term is similar to the second term but it represents the entirety of the adjunction arguments.

To provide a clear description, here is an example for the sentence in Figure 1.

$$\langle \text{提供}, (Arg0 : 1, Arg1 : 1, Arg2 : 1), (ArgM - Tmp : 1, ArgM - Adv : 1) \rangle$$

From the definition of PAC, all of the components of the predicate-argument structure are clearly divided into three parts: the predicate, the core arguments and the adjunction arguments. In PAC, the dependency between the predicate and the arguments is implied in n . The reason why we separate the core arguments from the adjunction arguments is that their functions in the sentence are different and their relationships to the predicate are also different. In fact, in the optimizing process, our model first acquires an optimal solution for the core part and the adjunction part respectively, and then combines the two parts into a whole one, which is the optimal solution for SRL.

3.2 Expression Ability of PAC

An eligible concept should embody many linguistic phenomena that are related to the predicate-argument structure. We declare that the Predicate-Argument-Coalition is easy and adequate to express some complex linguistic phenomena. Let us take the examples below to show the expression ability of PAC.

- a) 百分之九十五的 产品 销往 海外。
95% of the products were sold to abroad.
- b) 经营 成本 降到 最低。
Operation cost dropped to the minimum
- c) 国际油价 暴跌。
The global price of oil dropped sharply.
- d) 1994 年 墨西哥 金融 危机 爆发。
In 1994, Mexico's financial crisis broke out.

A common constraint in SRL system is no duplication for core arguments. To satisfy the constraint, the only necessary measure in PAC is that all of the numbers n in the ARGX multi-set are not allowed larger than one. In example (a), the predicate is ‘销往’ (be sold to) and according to the literal meaning of ‘销往’ (be sold to), there must be a “location” that indicates the place where products are sold to. Under this circumstance, the “location” is labeled ARG2. Thus, when the word ‘销往’ (be sold to) appears in a sentence, there must be a candidate that is labeled as ARG2. To handle this linguistic phenomenon, we only ensure that the n of ARG2 in the ARGX multi-set equals one. In examples (c) and (d), the predicates are intransitive. We can see that there is usually ARG0 or ARG1. We just keep the mutual exclusion of ARG0 and ARG1 in the ARGX multi-set. In other words, ARG0 and ARG1 never simultaneously emerge in a sentence with intransitive verbs. In this situation, we just keep the mutual exclusion of ARG0 and ARG1 in the ARGX multi-set.

The above examples and analysis prove much dependency between arguments and the predicate again, and PAC is easily used to represent the dependency. More importantly, we can obtain all predicates’ PAC statistics from training data without much labor.

4 Local Model

Following the traditional pipeline, we divide the SRL task into two phases – argument identification and argument classification. Before argument identification, candidate argument pruning is necessary due to a large number of candidates. Here, we implement a state-of-the-art pruning method as in Xue (2008).

4.1 Classifier

We adopt a discriminative model - Maximum Entropy model (ME), as our classifier because ME can easily be expanded to incorporate arbitrary features as a discriminative model.

4.2 Features

We need to train two classifiers: one for arguments identification and the other for arguments classification. The following lists the features that are utilized in the two classifiers.

The features used in arguments identification:

- Lexicalization features include the head word, the predicate and the predicate' verb class.
- Syntactic features include the head word's part-of-speech (POS), the path from the candidate argument to the predicate, and the path from the candidate argument to BA and BEI (Xue, 2008).
- Combination features include the predicate + head word, the predicate + syntactic tag of the candidate argument, the predicate' verb class + head word, the predicate's verb class + the syntactic tag of the candidate argument.

All of the above features are contained in the arguments classification. In addition, there are some other features:

- Position: the relative position of the candidate argument to the predicate.
- Subcat frame: the syntactic rule that expands the parent of the verb
- Phrase type: the syntactic tag of the candidate argument
- The first and the last word of the candidate argument
- Subcat frame+: the frame that consists of the NPs (Xue, 2008).

5 Proposed Global Generative Model

In Section 3, we propose a compact representation PAC for the predicate-argument structure. Based on PAC, we construct a generative model for SRL in this section.

5.1 Core Arguments and Free Arguments

Before our model's formulation, we take a second look at the arguments which are divided into the core arguments and the adjunction arguments. Here, we investigate their relationships with the predicate.

Core Arguments: These arguments are obviously sensitive with respect to the predicate. For example, the phrase '保險公司' (the insurance company) in Figure 1 is a core argument. If the predicate '提供' (provide) is not given, then it is difficult to label the argument ARG0.

Adjunction Arguments: In Figure 1, there are two adjunction arguments. The first argument is the phrase '截止目前' (until now), which has the label ARGM-TMP, which manifests the time of the word '提供' (provide); the second argument is the word '已' (has), which has the label ARGM-ADV, which is an adverbial modifier of the word '提供' (provide). For these two arguments, even though we do not know the predicate, we can classify the two arguments into ARGM-TMP and ARGM-ADV.

Based on the above observation, we make an assumption that adjunction arguments are independent from the predicate and we prefer calling these arguments Free Arguments (in the remaining parts, we keep calling them free arguments). Moreover, the independence assumption of the adjunction arguments is also helpful for simplifying our model.

Separate the Core Arguments from Free Arguments

Here we also implement an ME model to accomplish binary classification. The features include all of the argument classification features in the local model, and there are some additional features:

- The syntactic tag of the predicate’s parent
- The syntactic tag of the candidate argument’s parent
- The syntactic tag of the predicate

5.2 Formulations

Our model’s generative story is described as follows:

- 1) Generate a candidate PAC.
- 2) For a candidate PAC, one solution is obtained by assigning every label of the core-multiset and free-multiset of PAC to candidate arguments
- 3) Repeat (1) and (2).

We take SRL as the problem with structured output. The above generative process can be formulated as.

$$\begin{aligned} structure^* &= \arg \max P(structure \mid Cand, pred) \\ &= \arg \max P(structure \mid Cand, PAC)P(PAC \mid pred) \end{aligned}$$

in which, $Cand$ stands for all candidate arguments and consists of core arguments $Cand_{core}$ and free arguments $Cand_{free}$; $pred$ stands for the predicate.

Due to free arguments’ independence from the predicate, we use the classifier described in subsection 5.1 to separate core arguments from free argument. And then the deduction below is obtained:

$$\begin{aligned} structure^* & \\ &= \arg \max P(structure \mid PAC, Cand_{free})P(structure \mid PAC, Cand_{core})P(PAC \mid pred) \\ &= \arg \max P(structure_{free} \mid PAC_{free}, Cand_{free})P(structure_{core} \mid PAC_{core}, Cand_{core}) \\ &\quad P(PAC_{core}, PAC_{free} \mid pred) \\ &= \arg \max P(structure_{free} \mid PAC_{free}, Cand_{free})P(structure_{core} \mid PAC_{core}, Cand_{core}) \\ &\quad P(PAC_{core} \mid pred) \end{aligned}$$

In the above deduction, besides separating core candidates from free candidates, we also factor structure into $structure_{core}$ and $structure_{free}$, PAC into PAC_{core} and PAC_{free} .

5.3 Inference

There are three parts in the ultimate formulation. Here we give detailed description about how to infer the optimal solution.

The first part is to obtain $structure_{free}$ of the free arguments. Because there is no constraint about the free arguments in the optimization process, the optimization solution for $structure_{free}$ is exactly the same as that in the local model.

The second part is $P(structure_{core} | PAC_{core}, Cand_{core})$ is the probability of obtaining $structure_{free}$ given PAC_{core} and $Cand_{core}$ and also reflects the probability of assigning one tag of PAC_{core} to one candidate of $Cand_{core}$ without repetition. Since the PAC_{core} is a multi-set, we need to traverse all of the possible tag sequences to obtain the maximum. The detailed computation is given as follows:

$$P(structure_{core} | PAC_{core}, Cand_{core}) = \max_{\text{all sequences of } PAC_{core}} \prod p(Arg_i | cand_i)$$

in which $p(Arg_i | cand_i)$ is the probability of assigning the i -th tag of the sequence to the i -th candidate argument according to the local model.

The third part is $P(PAC_{core} | pred)$ is the probability of a candidate multi-set PAC_{core} given the predicate $pred$. The maximum likelihood estimation for $P(PAC_{core} | pred)$ can be computed by the following equation:

$$\begin{aligned} P(PAC_{core} | pred) \\ &= \frac{\text{count}(pred, PAC_{core})}{\sum \text{count}(pred, PAC'_{core})} \end{aligned}$$

In inferring $structure_{core}$, the second part stands for the local property while the third part reflects the inherent prior property of the predicate. The two parts provide a solution for $structure_{core}$. In summary, we obtain $structure_{free}$ through the first part and $structure_{core}$ through the second and the third part. After $structure_{free}$ and $structure_{core}$ are obtained, the two parts constitute a whole structure and the optimal $structure^*$ is our solution.

6 Experiments

6.1 Experiments Setup

We use Chinese Proposition Bank 1.0 in this experiment. According to the traditional division (Xue, 2008; Sun et al., 2009), all of the data are divided into three parts. 648 files (from `chtb_081.fid` to `chtb_899.fid`) are used as the training set. The second part includes 40 files from `chtb_041.fid` to `chtb_080.fid` as the development set. The test set is 72 files, which are `chtb_001.fid` to `chtb_040.fid` and `chtb_900.fid` to `chtb_931.fid`. We adopt the Berkeley parser to perform auto parsing for SRL and re-train the parser on the training set.

6.2 Results

We have compared our GGM with the local model in Table 1 and the evaluation criterion is F1. From Table 1, we can see that for the core arguments, our model significantly outperforms the local model by approximately 0.8 points. For free arguments, GGM’s score is higher than the baseline, which benefits from core and free arguments separation stage. The overall performance has been improved from 74.04 to 74.73.

To provide a further description, Table 2 lists the detailed numbers that are related to the F1 value. “False” means the number of null arguments that are distinguished as arguments. “Miss” means the number of missing arguments. “Right” and “Error” means the numbers of arguments that are classified correct and wrong respectively. Because both GGM and the local model take the same pruning and argument identification steps, the “False” number and the “Miss” number are the same. However, GGM’s error number declines by 13.8% compared with the local model.

Table 1. Comparison with the Local Model

	Num	Local	GGM
ARG0	2023	67.55	68.35
ARG1	2649	78.63	79.43
ARG2	359	62.79	65.41
ARG3	28	50.00	55.32
ARG4	5	54.55	72.73
ARGM	3023	74.81	75.10
all	8432	74.04	74.73

Table 2. Comparison on detailed “False”, “Right”, “Error” and “Miss” numbers with the Local Model

Method	False	Right	Error	Miss
Local	1159	5897	377	2208
GGM	1159	5932	322	2208

6.3 Advanced Features for Free Arguments

It is noted that in Table 1, the performance of free arguments is not improved as core arguments since our GGM focuses on core arguments. In [4], they thought that different features were necessary to capture the crucial information for classifying the core arguments and the free arguments. In their system, they first discriminate as to whether arguments belong to either core arguments or free arguments by a binary classifier, and then, they label core candidates and free candidates with different features. Motivated by their approach, we define a new feature called Word Semantic Class (WSC) for classifying free arguments.

In the ArgM-TMP arguments of PropBank, there must be a word that means the time and for the ArgM-LOC arguments, there must be a word that means the location. Because these words are usually nouns, we could easily extract the time and location words from the corpus, and these are two types of Word Semantic Classes. For the other free arguments, there are some exclusive words. For example, in ArgM-Dir, there is always a preposition that indicates a direction such as ‘朝着’ (face to). Moreover, these words often appear at the beginning of the argument. Thus, we can extract these exclusive words according to the first word of the arguments. Some typical examples of WSC are shown in Table 3.

Table 3. Examples of Word Semantic Class

	WSC
春天 (spring)	TMP
埃及 (Egypt)	LOC
为了 (to/for)	PRP
朝着 (face to)	DIR
但是 (but)	DIS
和 (and)	CRD
根据 (accord to)	MNR
即使 (although)	CND

We have evaluated the effect of the WSC feature for SRL. The results are shown in Table 4. We divide the local model’s classification features into two types: base features and pred features. The pred features are related to the predicate and the base features are the other features. We can see that if we only use the base features, the performance drops to 71.86 from 74.04. After the WSC features are added, the performance rises to 74.94 sharply. After we add the pred features into the model, the performance is improved slightly.

Table 4. Evaluation Results on the Feature WSC

	Base	+WSC	+Pred
GGM	71.86	74.94	75.01

6.4 Comparison with Other Methods

We also compared GGM with other methods and results are shown in Table 5. It can be seen that the approaches that incorporate argument dependencies (Toutanova et al., 2008; GGM; GGM+) are better than others. Meanwhile, our approach outperforms the state-of-the-art method by 0.5 F1 points. It is generally believed that Chinese language generation is more complex and arbitrary than English. Therefore, the sequence structure in Toutanova’s joint model is not adequate to catch the global properties of the candidate arguments while our PAC can handle these in an effective way.

Table 5. Comparison with Other Methods. GGM+ means adding the new feature WSC into GGM

Methods	F ₁
Xue(2008)	71.9
Sun(2009)	74.12
Toutanova(2008)	74.50
GGM	74.73
GGM+	75.01

7 Conclusion and Future Work

In this paper, to address the argument dependencies in SRL, we propose a global generative model in which a novel concept Predicate-Argument-Coalition is defined to represent the predicate-argument structure. The existing approaches treat the predicate-argument structure as a sequence structure but in some languages such as Chinese, the sequence structure is not adequate to describe a complex predicate-argument structure. Unlike them, our model can effectively mine the inherent properties of the predicates through PAC, which is helpful for SRL. Experiment results on Chinese PropBank and English PropBank demonstrate the superiority of our approach.

Acknowledgments. We thank the three anonymous reviewers for their helpful comments and suggestions. The research work has been partially funded by the Natural Science Foundation of China under Grant No.61333018 and the International Science & Technology Cooperation Program of China under Grant No.2014DFA11350, and also the High New Technology Research and Development Program of Xinjiang Uyghur Autonomous Region under Grant No.201312103 as well.

References

1. Baker, C., Fillmore, C., Lowe, J.: The berkeley framenet project. In: Proceedings of COLING-ACL 1998 (1998)
2. Cohn, T., Blunsom, P.: Semantic Role Labeling with Tree Conditional Random Fields. In: Proceedings of CONLL 2005 (2005)
3. Christensen, J., Mausam, Soderland, S., Etzioni, O.: Semantic Role Labeling for Open Information Extraction. In: Proceedings of ACL 2010 (2010)
4. Ding, W., Chang, B.: Improving Chinese Semantic Role Classification with Hierarchical Feature Selection Strategy. In: Proceedings of EMNLP 2008 (2008)
5. Gildea, D., Jurafsky, D.: Automatic labeling for semantic roles. *Computational Linguistics* 28(3), 245–288 (2002)
6. Jiang, Z., Li, J., Ng, H.T.: Semantic Argument Classification Exploiting Argument Interdependence. In: Proceedings of IJCAI 2005 (2005)
7. Kingsbury, P., Palmer, M.: From Treebank to PropBank. In: Proceedings of LREC 2002 (2002)

8. Li, J., Zhou, G., Ng, H.T.: Joint Syntactic and Semantic Parsing of Chinese. In: Proceedings of ACL 2010 (2010)
9. Liu, D., Gildea, D.: Semantic role features for machine translation. In: Proceedings of COLING 2010 (2006)
10. Moschitti, A., Pighin, D., Basili, R.: Semantic role labeling via tree kernel joint inference. In: Proceedings of CONLL 2006 (2006)
11. Moschitti, A., Pighin, D., Basili, R.: Tree Kernels for Semantic Role Labeling. *Computational Linguistics* 34(2), 193–224 (2008)
12. Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., Doi, H.: Semantic Role Labeling Using Support Vector Machines. In: Proceedings of CONLL 2005 (2005)
13. Narayanan, S., Harabagiu, S.: Question Answering based on Semantic Structures. In: Proceedings of COLING 2004 (2004)
14. Punyakanok, V., Roth, D., Yih, W., Zimak, D.: Semantic Role Labeling via Integer Linear Programming Inference. In: Proceedings of COLING 2004 (2004)
15. Surdeanu, M., Harabagiu, S., Williams, J., Aarseth, P.: Using Predicate-Argument Structures for Information Extraction. In: Proceedings of ACL 2003 (2003)
16. Surdeanu, M., Turmo, J.: Semantic Role Labeling Using Complete Syntactic Analysis. In: Proceedings of CONLL 2005 (2005)
17. Sun, W., Sui, Z., Wang, M., Wang, X.: Chinese Semantic Role Labeling with Shallow Parsing. In: Proceedings of ACL 2009(2009)
18. Toutanova, K., Haghghi, A., Manning, C.D.: Joint learning improves semantic role labeling. In: Proceedings of ACL 2005(2005)
19. Toutanova, K., Haghghi, A., Manning, C.D.: A Global Joint Model for Semantic Role Labeling. *Computational Linguistics* 34(2), 161–191 (2008)
20. Tsai, T.-H., Wu, C.-W., Lin, Y.-C., Hsu, W.-L.: Exploiting Full Parsing Information to Label Semantic Roles Using an Ensemble of ME and SVM via Integer Linear Programming. In: Proceedings of CONLL 2005 (2005)
21. Wu, D., Fung, P.: Can semantic role labeling improve smt. In: Proceedings of EAMT 2009 (2009)
22. Wu, S., Palmer, M.: Semantic mapping using automatic word alignment and semantic role labeling. In: Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation (2011)
23. Deyi, X., Min, Z., Haizhou, L.: Modeling the Translation of Predicate-Argument Structure for SMT. In: Proceedings of ACL 2012 (2012)
24. Xue, N., Palmer, M.: Annotating the Propositions in the Penn Chinese Treebank. In: The Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (2003)
25. Xue, N., Palmer, M.: Calibrating Features for Semantic Role Labeling. In: Proceedings of EMNLP 2004 (2004)
26. Xue, N.: Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics* 34(2), 225–255 (2008)
27. Yi, S., Palmer, M.: The Integration of Syntactic Parsing and Semantic Role Labeling. In: Proceedings of CONLL 2005 (2005)
28. Zhai, F., Zhang, J., Zhou, Y., Zong, C.: Machine Translation by Modeling Predicate-Argument Structure Transformation. In: Proceedings of COLING 2012 (2012)
29. Tao, Z., Zong, C.: A Minimum Error Weighting Combination Strategy for Chinese Semantic Role Labeling. In: Proceedings of EMNLP 2010 (2010)

Chinese Comma Disambiguation on K-best Parse Trees

Fang Kong and Guodong Zhou

School of Computer Science and Technology
Soochow University, China
{kongfang, gdzhou}@suda.edu.cn

Abstract. Chinese comma disambiguation plays key role in many natural language processing (NLP) tasks. This paper proposes a joint approach combining K-best parse trees to Chinese comma disambiguation to reduce the dependent on syntactic parsing. Experimental results on a Chinese comma corpus show that the proposed approach significantly outperform the baseline system. To our best knowledge, this is the first work improving the performance of Chinese comma disambiguation on K-best parse trees. Moreover, we release a Chinese comma corpus which adds a layer of annotation to the manually-parsed sentences in the CTB (Chinese Treebank) 6.0 corpus.

Keywords: Chinese Comma Disambiguation, Discourse Analysis, Sentence Segmentation, K-best parse trees.

1 Introduction

The Chinese commas function quite different from its English counterpart. On one hand, they can signal the sentence boundary. Similar to English, Chinese also uses periods, question marks, and exclamation marks to indicate sentence boundaries. Where these punctuation marks exist, sentence boundaries can be determined unambiguously. The difference is that the Chinese commas, as the most common form of punctuation, can also function similarly as the English periods in many kinds of contexts. On the other hand, the Chinese commas can also signal the boundary of discourse units and anchor discourse relations between text spans. Observing the CTB 6.0 corpus, we find that implicit discourse relations occupy more than 75% and much of them are anchored by the Chinese commas.

In recent years, Chinese comma disambiguation has attracted increasing attention due to its importance in many NLP tasks, such as sentence segmentation ([4,6,9,5]) and discourse analysis ([10,8]). Previous work has achieved reasonable success, they also found that the performance of Chinese comma disambiguation heavily depended on the performance of syntactic parser. In this paper, we classify the Chinese commas into seven categories based on syntactic patterns and annotate a Chinese comma corpus which adds a layer of annotation to the manually-parsed sentences in the CTB 6.0 corpus. Using the annotated corpus,

a machine learning approach to Chinese comma disambiguation is proposed. Finally, a joint approach based on K-best parse trees is employed to reduce the dependent on syntactic parsing. Experiments on our Chinese comma corpus show that our joint approach can significantly improve the performance of Chinese comma disambiguation with automatic parse trees.

The rest of this paper is organized as follows. Section 2 introduces the related work from syntactic parsing and discourse analysis. In Section 3, we present our Chinese comma classification scheme and briefly overview our annotated Chinese comma corpus. Section 4 describes a machine learning approach to Chinese comma disambiguation as a baseline. In Section 5, a joint approach combining k-best syntactic parse trees is proposed. Section 6 presents the experiments and results. Finally, we give the conclusion and further work in Section 7.

2 Related Work

The Chinese comma can not only function similarly as the English periods, but also act as the boundary of sentences or discourse units. Currently, many research work about Chinese comma disambiguation has been conducted from the perspective of sentence segmentation and discourse analysis.

For Chinese sentence segmentation, the related work can be classified into two categories: in the context of syntactic parsing for long sentences, and serving for some NLP applications such as machine translation and empty category recovery. The representative work includes: Jin et al. [4] and Li et al.[6] view Chinese comma disambiguation as a part of a “divide-and-conquer” strategy to syntactic parsing. Long sentences are split into shorter sentence segments on commas before they are parsed, and the syntactic parses for the shorter sentence segments are then assembled into the syntactic parse for the original sentence. Xue and Yang [9] view Chinese comma disambiguation as the detection of loosely coordinated clauses separated by commas, which are syntactically and semantically complete on their own and do not have a close syntactic relation with one another. In this way, some downstream tasks such as parsing and Machine Translation can be simplified. Kong and Zhou [5] employ a comma disambiguation method to improve syntactic parsing and help determine clauses in Chinese. Based on the detected clauses, a clause-level hybrid approach is proposed to address specific problems in Chinese empty category recovery and achieves significant performance improvement.

For discourse analysis, related work find that the Chinese comma can be further viewed as a delimiter of elementary discourse units (EDUs) and the anchor of discourse relations. Disambiguating the comma is thus necessary for the purpose of discourse segmentation, the identification of EDUs, a first step in building up the discourse structure of a Chinese text. The representative work includes: Yang and Xue [10] propose a discourse structure-oriented classification of the comma and conduct experiments with two supervised learning methods that automatically disambiguate the Chinese comma based on this classification. Motivated by the work of Yang and Xue, Xu et al. [8] also divide the Chinese

commas into seven categories based on syntactic patterns and propose three different machine learning methods to automatically disambiguate the Chinese commas.

All the previous work shows that the performance of Chinese comma disambiguation heavily depends on the performance of syntactic parser. Similar to the work of Yang and Xue [10] and Xue et al. [8], in this paper, we also classify the Chinese commas into seven categories based on syntactic patterns. Then a traditional machine learning approach will be employed to do Chinese comma disambiguation automatically. Based on this baseline system, we focus on improving the performance of Chinese comma disambiguation on K-best parse trees.

3 Chinese Comma Classification

Just as Zhou and Xue [11] noted, despite similarities in discourse features between Chinese and English, there are differences that have a significant impact on how discourse relations could be best annotated. For example, as illustrated in (1), there are six commas. In its corresponding English translation, we only can find the first, fifth and sixth commas. The second comma does not mark the boundary of discourse unit and is not translated, the third one corresponds to an English period. And the fourth comma marks the boundary of discourse unit but not translated in English. In fact, the Chinese sentence can be split into five discourse units marked with (a)–(e).

(1) 对此, [1]

[(a) 浦东不是简单的采取“干一段时间, [2]等积累了经验以后再制定法规条例”的做法, [3]]

[(b) 而是借鉴发达国家和深圳等特区的经验教训, [4]]

[(c) 聘请国内外有关专家学者, [5]]

[(d) 积极、及时地制定和推出法规性文件, [6]]

[(e) 使这些经济活动一出现就被纳入法制轨道]。

“In response to this ,[1]

[(a) Pudong is not simply adopting an approach of ” work for a short time and then draw up laws and regulations only after waiting until experience has been accumulated . ”]

[(b) Instead , Pudong is taking advantage of the lessons from experience of developed countries and special regions such as Shenzhen]

[(c) by hiring appropriate domestic and foreign specialists and scholars ,[5]]

[(d) by actively and promptly formulating and issuing regulatory documents ,[6]]

[(e) and by ensuring that these economic activities are incorporated into the sphere of influence of the legal system as soon as they appear .]”

Figure 1 and Figure 2 show the corresponding syntactic parse tree and discourse parse tree, respectively. From the figures, we can find that different commas with different syntactic patterns can function differently in discourse modeling.

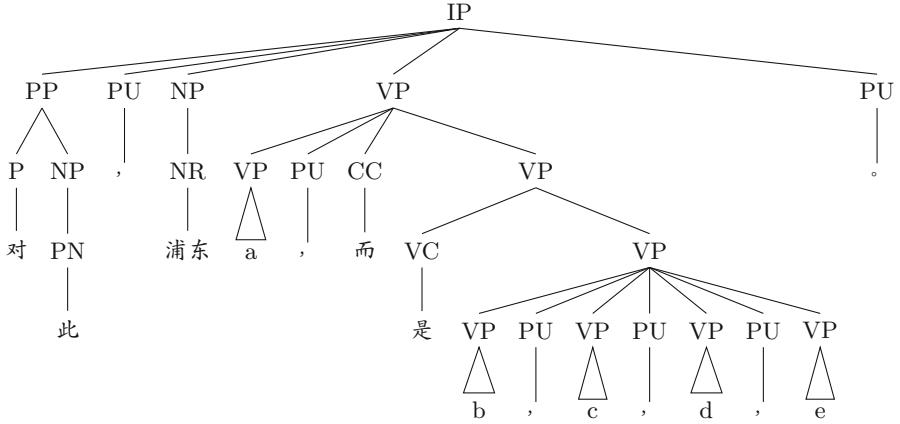


Fig. 1. Syntactic parse tree corresponding to Example (1)

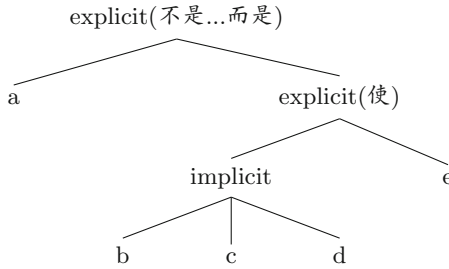


Fig. 2. Discourse parse tree corresponding to Example (1)

Referring the work of Yang and Xue [10] and Xu et al. [8], we classify the Chinese comma into seven hierarchically organized categories based on the syntactic patterns. Figure 3 shows our classification scheme. The description of the categories are following:

- SB, sentence boundary. The loosely coordinated IPs that are the immediate children of the root IP to be independent sentences, and the commas separating them to be delimiters of sentence boundary.
- COORDIP, coordinated IPs that are not the immediate children of the root IP are also considered to be discourse units and the commas linking them are labeled COORDIP.
- COORDVP, coordinated VPs, when separated by the comma, are not semantically different from coordinated IPs. The only difference is that the coordinated VPs share a subject.

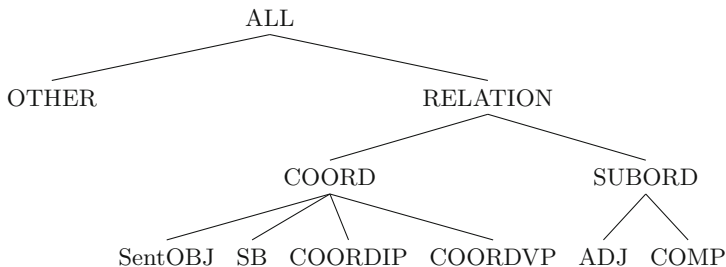


Fig. 3. Chinese Comma Classification

- SentOBJ, this category is for commas linking two coordinated IPs in the object phrase.
- COMP, when a comma separates a verb governor and its complement clause, this verb and its subject generally describe the attribution of the complement clause.
- ADJ, this category is for commas linking a subordinate clause with its main clause.
- OTHER, the remaining cases of comma.

Knowing the above Chinese comma classification scheme, all the commas in the CTB can be mapped to one of the seven classes based on the syntactic patterns. Using semi-automatic way (i.e., human adjust after rule-based approach), we build a Chinese comma corpus adding a layer of comma annotation in the CTB 6 corpus. Table 1 shows the distribution of all the comma instances over the seven categories.

Table 1. The distribution of the comma instance over different categories

Category	Numbers	Percent(%)
SB	13215	25.5
COORDIP	552	1.1
COORDVP	5790	11.2
SentOBJ	2051	4
COMP	3274	6.3
ADJ	2347	4.5
OTHER	24675	47.5
Overall	51886	100

4 Baseline System: A Maximum Entropy Approach

After the commas are labeled, we have basically turned comma disambiguation into multiple classification problem. We trained a Maximum Entropy classifier with the Mellet machine learning package¹ to classify the Chinese commas.

¹ <http://mallet.cs.umass.edu>

The features are extracted from gold standard parse trees and automatical parse trees, respectively. We implement features described in Xue and Yang [9], and also introduce a set of new features. Table 2 lists the new features employed in Chinese comma classification, which reflect the properties of the context where current comma occurs. The third column of Table 2 shows the features corresponding to Figure 1, considering the fourth comma between span c and d as the current comma in question.

Table 2. New features employed in comma classification

Num	Description	Example
1	Conjunction of the siblings of the comma	VP+VP-PU-VP-PU-VP
2	Conjunction of the siblings of the comma ' s parent node	VC-VP
3	Whether the parent of the comma is a coordinating VP construction. A coordinating VP construction is a VP that dominates a list of coordinated VPs	True
4	Whether the Part-of-speech tag of the leftmost sibling of the comma ' s parent node is a PP construction	False
5	Whether the siblings of the comma ' s parent node has and only has an IP construction	False
6	Whether the first leaf node ' s Part-of-speech tag of the comma ' s parent node is CS or AD construction	False
7	Whether the right siblings of the comma has the NP+VP construction	False
8	Whether the first child of the comma ' s left sibling is the PP construction	False
9	If the leftmost sibling of the comma is an IP construction, whether the first child of the comma ' s right sibling is the CS or AD construction	False

5 Refined System: K-best Combination Approach

Note that most of features employed in our baseline system are extracted from syntactic parse trees. Consistent with previous work on Chinese comma disambiguation, the performance of our baseline system should heavily depend on the performance of syntactic parser. In this section, we will propose a k-best combination approach to address this problem.

As well known, parsing re-ranking has been shown to be an effective technique to improve parsing performance [2,1,3]. This technique uses a set of linguistic features to re-rank the k-best output on the forest level or tree level. Motivated by these work, using the general framework of re-ranking, we joint Chinese comma disambiguation with the selection of the best parse tree. The idea behind this approach is that it allows uncertainty about syntactic parsing to be carried forward through a K-best list, and that a reliable comma disambiguation system, to a certain extent, can reflect qualities of syntactic parse trees. Given a sentence s , a joint parsing model is defined over a comma c and a parse tree t in a log-linear way:

$$Score(c, t|s) = (1 - \alpha) \log P(c|t, s) + \alpha \log P(t|s)$$

while $P(t|s)$ is returned by a probabilistic syntactic parsing model, and $P(c|t, s)$ is returned by a probabilistic comma classifier. In our K-best combination approach, $P(t|s)$ is calculated as the product of all involved decisions' probabilities in the syntactic parsing model, and $P(c|t, s)$ is calculated as the product of all the commas' probabilities in a sentence. Here, the parameter α is a balance factor indicating the importance of the comma disambiguation model.

In particular, (c^*, t^*) with maximal $Score(c, t|s)$ is selected as the final syntactic parsing tree and the comma disambiguation result.

6 Experimentation

6.1 Experimental Settings

We use the CTB 6.0 in our experiments and divide it into training, development and test sets, as shown in Table 3. All our classifiers are trained using the the Mallet machine learning package² with the default parameters (i.e. without smoothing and with 100 iterations). Under the automatic setting, the Berkeley parser [7] is used to generate top-best parse trees and 50-best parse trees, respectively.

Table 3. CTB 6 Data set division

Data	File ID
Train	81-325,400-454,500-554,590-596,600-885,1001-1017,1019,1021-1035,1037-1043,1045-1059,1062-1071,1073-1078,1100-1117,1130-1131,1133-1140,1143-1147,1149-1151
Dev	41-80,1120-1129,2140-2159,2280-2294,2550-2569,2775-2799,3080-3109
Test	1-40, 901-931,1018,1020,1036-1044,1060-1061,1072, 1118-1119,1132,1141-1142,1148

6.2 Results

Table 4 lists the results under three different settings: using gold standard parse trees, using top-best parse trees, and using 50-best parse trees.

The second column shows the results of our comma disambiguation system using gold standard parse trees. From the results, we can find that our baseline system performs best on the category COMP. Both precision and recall are more than 95%, and the F-score is 97.81%. While on the category COORDIP, our system achieves only 50.0% in F1-measure much due to the poor recall. The overall accuracy of our comma disambiguation system using gold parse trees is 87.76%.

² <http://mallet.cs.umass.edu>

The third column shows the performs of our comma disambiguation system using top-best automatic parse trees. For the category COMP, the system also achieves satisfactory results. But for the category COORDIP and the category ADJ, our system only achieves about 28% in F1-measure. In comparison with using gold standard parse trees, the performance of every category is reduced. Especially for the category SentOBJ and the category ADJ, the F-score reduced by about 28% and 38%, respectively. The overall accuracy reduced about 5%.

The fourth column shows the results of our comma disambiguation system joint with 50-best parse trees. In comparison with using top-best parse trees, we can find that the refined system can achieve better performance on all the categories except the category COORDVP. Except the category COMP and OTHER, the improvement on every category is larger than 10% in F1-measure. And the overall accuracy of the refined system improves about 1.5% comparing with using top-best parse trees.

Although our refined system reduces the performance gap between using automatic parse trees and using gold parse trees by about 30%, it still lags behind using gold standard parse trees about 3.7% in overall accuracy. This suggests that there exists some room in the performance improvement for the joint mechanism with K-best parse trees.

Table 4. Overall accuracy as well as the results for each individual category

	standard parse trees			top-best parse trees			50-best parse trees		
	P	R	F	P	R	F	P	R	F
SB	62.16	88.46	73.02	55.56	76.92	64.52	63.89	88.46	74.19
COORDIP	100.0	33.33	50.0	100	16.17	28.57	100.0	33.33	50.0
COORDVP	84.85	72.73	78.32	77.92	77.92	77.92	74.67	72.73	73.68
SentOBJ	80.95	94.44	87.18	50.0	72.22	59.09	60.0	83.33	69.77
COMP	100.0	95.71	97.81	98.46	91.43	94.81	95.71	95.71	95.71
ADJ	66.67	66.67	66.67	25.0	33.33	28.57	100.0	33.33	50.0
OTHER	89.87	91.42	90.64	88.39	84.98	86.65	89.29	85.84	87.53
Overall(Acc)	87.76			82.45			84.06		

7 Conclusion and Future Work

Based on syntactic patterns, we classify the Chinese commas into seven categories and annotate a Chinese comma corpus adding a layer of annotation in the CTB 6.0 corpus. Using this annotated corpus, we propose a approach to disambiguate the Chinese commas as a first step toward discourse analysis. In order to reduce the dependent on syntactic parsing, a joint mechanism based on K-best parse trees is proposed. Experiment results show the effectiveness of our joint approach.

In our future work, we will find more effective joint inference mechanism to improve the performance of Chinese comma disambiguation.

Acknowledgements. This research is supported by Key project 61333018 and 61331011 under the National Natural Science Foundation of China, Project 6127320 and 61472264 under the National Natural Science Foundation of China, Project 2012AA011102 under the National 863 Program of China, and Project 11KJA520003 under the Natural Science Major Fundamental Research Program of the Jiangsu Higher Education Institutions.

References

1. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and maxent discriminative reranking. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 173–180. Association for Computational Linguistics, Ann Arbor (2005), <http://www.aclweb.org/anthology/P05-1022>
2. Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pp. 263–270. Association for Computational Linguistics, Philadelphia (2002), <http://www.aclweb.org/anthology/P02-1034>
3. Huang, L.: Forest reranking: Discriminative parsing with non-local features. In: Proceedings of ACL 2008: HLT, pp. 586–594. Association for Computational Linguistics, Columbus (2008), <http://www.aclweb.org/anthology/P/P08/P08-1067>
4. Jin, M., Kim, M.Y., Kim, D., Lee, J.H.: Segmentation of chinese long sentences using commas. In: Streiter, O., Lu, Q. (eds.) ACL SIGHAN Workshop 2004, pp. 1–8. Association for Computational Linguistics, Barcelona (2004)
5. Kong, F., Zhou, G.: A clause-level hybrid approach to chinese empty element recovery. In: IJCAI. IJCAI/AAAI (2013)
6. Li, X., Zong, C., Hu, R.: A hierarchical parsing approach with punctuation processing for long chinese sentences. In: Proceeding of the Second International Joint Conference on Natural Language Processing: Companion Volume to the Proceedings of Conference Including Posters/Demos and Tutorial Abstracts, pp. 17–24 (2005), <http://anthology.aclweb.org/I/I05/I05-2002>
7. Petrov, S., Klein, D.: Improved inference for unlexicalized parsing. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 404–411. Association for Computational Linguistics, Rochester (2007), <http://www.aclweb.org/anthology/N/N07/N07-1051>
8. Xu, S., Li, P.: Recognizing chinese elementary discourse unit on comma. In: IALP, pp. 3–6. IEEE (2013)
9. Xue, N., Yang, Y.: Chinese sentence segmentation as comma classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 631–635. Association for Computational Linguistics, Portland (2011), <http://www.aclweb.org/anthology/P11-2111>

10. Yang, Y., Xue, N.: Chinese comma disambiguation for discourse analysis. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 786–794. Association for Computational Linguistics, Jeju (2012), <http://www.aclweb.org/anthology/P12-1083>
11. Zhou, Y., Xue, N.: Pdtb-style discourse annotation of chinese text. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 69–77. Association for Computational Linguistics, Jeju Island (2012), <http://www.aclweb.org/anthology/P12-1008>

Event Schema Induction Based on Relational Co-occurrence over Multiple Documents

Tingsong Jiang, Lei Sha, and Zhifang Sui

Key Laboratory of Computational Linguistics, Ministry of Education,
Institute of Computational Linguistics,
School of Electronics and Engineering and Computer Science,
Peking University, Beijing, China
{tingsong, shalei, szf}@pku.edu.cn

Abstract. Event schema which comprises a set of related events and participants is of great importance with the development of information extraction (IE) and inducing event schema is prerequisite for IE and natural language generation. Event schema and slots are usually designed manually for traditional IE tasks. Methods for inducing event schemas automatically have been proposed recently. One of the fundamental assumptions in event schema induction is that related events tend to appear together to describe a scenario in natural-language discourse, meanwhile previous work only focused on co-occurrence in one document. We find that semantically typed relational tuples co-occurrence over multiple documents is helpful to construct event schema. We exploit the relational tuples co-occurrence over multiple documents by locating the key tuple and counting relational tuples, and build a co-occurrence graph which takes account of co-occurrence information over multiple documents. Experiments show that co-occurrence information over multiple documents can help to combine similar elements of event schema as well as to alleviate incoherence problems.

Keywords: event schema, information extraction, co-occurrence analysis.

1 Introduction

Event schema is a template which comprises a set of events with prototypical transitions as well as a set of slots or roles representing the participants. Various roles or slots are contained in one event, such as the perpetrator, victim, and instrument in a bombing event. Event schema is helpful for information extraction and other NLP tasks due to the fact that both abstraction and concreteness are necessary information for people to understand the event. Traditional IE systems design the template manually and focus on extracting structured information concerning events to fill predefined templates. This approach often limits event templates' range of application to a relatively narrow area and is only applied in a particular domain.

One of the basic assumptions in event schema induction is that related events tend to appear together to describe a scenario in natural-language discourse, meanwhile previous work only focused on co-occurrence in one document. Besides, using relational tuples and generalization also makes the schema loose and scattered. We find that semantically typed relational tuples co-occurrence over multiple documents is helpful to construct event schema.

First, relational tuples co-occurrence over multiple documents may help to combine some loose and relative event schemas. Some event schemas may be scattered due to their lack of enough co-occurrence, such as the events shown in Figure 1. The event schemas in Figure 1 make some assertions that a person returning to his work but the schemas are somehow similar and describe the event parallelly which seems not that simple. If we exploit the relational co-occurrence over multiple documents, we can find that schema 1 in document A and schema 2 in document B have something in common where they tell us a person returned to some location at some time. We notice that it is induced from a document that describes the scientist Schwarzschild and his life in war, both schema 1 and schema 2 have the tuple that (Schwarzschild, return in, United States Army), and if we find the co-occurrence in two documents and count the relational tuples again, we can combine the event schemas and construct a relatively more abstract and complete event schema.

Schema 1:			
A1:[person]	return to	A0: [none]	lineup
A1:[person]	return against	A3:[location;organization]	
A1:[person]	return for	A4:[activity;game]	
A1:[person]	return after	A6: [none]	absence
A1:[person]	return in	A8:[location] United States Army	
A1:[person]	return until	A9:[time_unit]	
Schema 2:			
A0:[person]	return as	A1:[person]	
A0:[person]	return after	A5: [none]	year
A0:[person]	return to	A9:[time_period]	
A0:[person]	be survive by	A10: [none]wife	
A0:[person]	return in	A11:[location] United States Army	
Schema 3:			
A0:[person]	leave for	A1:[location] Europe	
A0:[person]	leave with	A6:[person]	
A0:[person]	leave to meet with	A8:[person]	
A0:[person]	leave after	A9: [none]	season
A0:[person]	leave on	A11:[organization]	
A0:[person]	leave as	A12:[leader]	

Fig. 1. An example of Niranjan, Stephen and Mausam and Oren(2013)'s system. Schema 1~3 are similar and describe the same event that one scientist returned to his work or location and can be combined and integrated.

Furthermore, we find in the experiment that relational tuples co-occurrence over multiple documents may also help alleviate incoherence problems. Chamber's system lack coherence because of the representation that uses pairs of elements from an assertion, thus, treating subject-verb and verb-object separately, and in a result their system may mix unrelated events and have roles whose entities do not play the same role in the schema. For example Niranjan pointed out Chamber's system mixed the events of fire spreading and disease spreading in Niranjan, Stephen and Mausam and Oren(2013), and they used relational tuples instead. However, we find that incoherence problems still exists.

In this paper we propose a method for event schema induction based on relational co-occurrence over multiple documents which overcomes scatter problems and incoherence in event schema induction. The rest of the paper is organized as follows: in Section 2 we review the related work while Section 3 presents a general overview of our approach and how relational co-occurrence over multiple documents is obtained and used in event schema induction. Finally, Section 4 gives the results of our approach.

2 Related Work

Since the late 1980s, information extraction began to flourish, this is mainly attributed to Message Understanding Conference (MUC) and Automatic Content Extraction (ACE) meeting. Event template extraction has been explored in the MUC-4 scenario template task. This task concentrated on pipeline models which decouple the task into the sub-tasks of field extraction and event-based text segmentation. The application of the current event schema generation and other semantic processing tasks is mainly by manual effort to define the target representation and annotate the examples to train the machine-learning system, besides MUC template is confined to specific areas, so large-scale event schema extraction system is desperately needed.

Event schemas in NLP were not automatically induced until the seminal work of Chambers and Jurafsky(2009). Their work is fully automatical and can deal with large text corpora. Early work by Chambers and Jurafsky (2008; 2009) showed that we could exploit the event sequences from text to induce event schemas automatically. They used pair-wise subject-verb and verb-object representation which may result to incoherence in the real world. Chambers and Jurafsky (2011) had applied unsupervised techniques to combine clustering, semantic roles, and syntactic relations so as to construct templates as well as to fill slots. Niranjan, Stephen, Mausam and Oren (2013) extended the system using relational n-grams to find the co-occurrence statistics of relational subject-verb-object tuples instead of pair-wise subject-verb/verb-object representation, and they aimed to overcome incoherence problems. However, Niranjan's system only considered the co-occurrence statistics in one document which may omit relations and information over multiple documents.

3 Event Schema Induction Based on Co-occurrence over Multiple Documents

In this section, we show the event schema induction process and deal with acquisition of the relational tuples co-occurrence over multiple documents.

3.1 System Overview

The overall system is based on the idea that both frequently co-occurring relational tuples in one document and over different documents can reflect the relatedness of assertions about real-world events. The preprocessing includes extracting the relational tuples from the corpus and counting the co-occurrence of relational tuples. The relational tuples can be treated as the nodes of a graph, and we can apply graph analysis to find the most relational nodes and then we can cluster them as part of the arguments of the event schema.

Event schema induction of Chamber’s work is focused on the verbs and their attribute thus verbs are the main features that distinguish one event from others according to the relations between verbs and events. Niranjana, Stephen, Mausam and Oren (2013) proposed that relational tuples is more expressive. We use Open Information Extraction (Mausam et al.,2012) and extract the relational tuples of the form (Arg1 , Relation , Arg2) which may contain more coherence than pair-wise representation. To reduce the sparsity resulted from tuples we use head verb and preposition to describe a phrase.

To ensure the tuples are not overly specific it can be generalized by semantic types from WordNet(Niranjana, Stephen, Mausam and Oren(2013)). The set of types are: person, organization, location, time unit, number, amount, group, business, executive, leader, effect, activity, game, sport, device, equipment, structure, building, substance, nutrient, drug, illness, organ, animal, bird, fish, art, book, and publication. We use Stanford Named Entity Recognizer (Finkel et al., 2005), and also look up the argument in WordNet 2.1 and record the first three senses if they map to the target semantic types. The sentence “Woz and Jobs started Apple in my parents’ garage in 1976.” could be divided into tuples as:

- 1.(Wos and Jobs, started, Apple)
- 2.(Wos and Jobs, started in, my parents’ grage)
- 3.(Wos and Jobs, started in, 1976)

Then it is generalized as follows:

- 1.(Wos and Jobs, started, Apple)
- 2.<person>, start, <org>
- 3.<person>, start in, <location>
- 4.<person>, start in, <time_unit>

...

After relation extraction and representation, we calculate the co-occurrence tuples both in one document and over multiple documents as detailed in Section 3.2. In Section 3.3 we construct a relational graph using the relational tuples as nodes and

co-occurrence quality as weighed edges, and apply graph analysis to find the key elements for event schema.

3.2 Relational Tuples Co-occurrence over Multiple Documents

As mentioned above, considering co-occurrence over multiple documents may help to combine event elements and alleviate incoherence problem in event schema induction. In the corpus, some documents may refer to the same topic or implicate the relations, others just have nothing to do with it. Since the text corpus may be large, we should first cluster the documents according to their types and features, then we find the common tuples over two documents and count the co-occurrence.

3.2.1 An Acquisition Method for Relational Tuples Co-occurrence over Multiple Documents Based on Transitivity

The basic intuition is that most frequent co-occurrence of two tuples reflect tight connection in the real world. Co-occurrence in one document can be defined easily as the count that they appear in a window. But co-occurrence over multiple documents may be not that clear and easy. We find that co-occurrence over multiple documents may show transitivity. We consider the example of a bombing event. In the first news article, we calculate the co-occurrence number and find tuple X: (bomb, explode, <location>) and tuple Y: (bomb, kill, <person>) are highly connected. In the second news article we calculate tuple Y: (bomb, kill, <person>) and tuple Z: (<person>, be identified, perpetrator) are frequently appeared. We can infer that X and Z have a latent relation and we thus try to find the co-occurrence statistics between them.

If we find that X follows Y and the weighted co-occurrence reaches the threshold in one document, besides, we find that Z follows Y and also the weighted co-occurrence reaches the threshold. Then we can guess X and Z are relational somehow, and we should count the co-occurrence between them by comparing the two documents. Specifically, we should first find the same tuple that joins the two documents together, and then we add the distance together for one occurrence. By counting the co-occurrence number with their distances like this, a new “co-occurrence count” parameter is obtained. The database thus can be shown in Figure 2. Suppose in document A: a tuple list contains: id23=(bomb, explode in, <location>), id69=(bomb, kill, <person>), they occur once in distance=1 and they occur 27 times in the overall documents; In document B: the tuple list is in time order, i.e. id69= (bomb, kill, <person >), id71=(<person>, be sent to, hospital), id95= (<person>, be identified, perpetrator), id69 and id95 occur once in distance=2; then we consider the co-occurrence over these two documents, and find they occur once in distance=3. Likewise, we then count the co-occurrence of (id23, id95) in distance=3 over the cluster of documents and add up them to the table and get Count=47. Note that the “Count=47” is normalized to be fair to traditional co-occurrence in one document.

Table 1.

id	Arg1	Relation	Arg2	Count
...
23	Bomb	Explode in	<location>	547
24	Bomb	Explode in	Baghdad	22
25	Bomb	Explode in	Market	7
...
69	Bomb	Kill	<person>	173
...
<u>95</u>	<person>	Be identified	<org>	231
...

Table 2.

X	Y	flag	Distance	Count	E11	E12	E21	E22
...
23	<u>69</u>	0	1	27	25	0	0	0
23	<u>69</u>	0	2	35	33	0	0	0
...
23	<u>69</u>	0	10	62	59	0	0	0
<u>69</u>	23	0	1	6	0	0	0	0
...
<u>69</u>	<u>95</u>	0	1	18	0	0	32	0
<u>69</u>	<u>95</u>	0	2	12	0	0	9	0
...
<u>69</u>	<u>95</u>	0	10	54	0	0	50	0
<u>95</u>	<u>69</u>	0	1	14	0	36	0	0
...
23	<u>95</u>	1	3	47	0	0	0	0
23	<u>95</u>	1	4	39	0	0	0	0
...

Fig. 2. Table A represents the basic statistics of relational tuples count occur over the documents. Table B represents the co-occurrence statistics of relational tuples both in one documents and over multiple documents within different distances. Tag “flag=1” represents co-occurrence over multiple documents and “flag=0” means co-occurrence in each document and we add up each document’s statistics to fill the table. Parameter E11 means X.Arg1=Y.Arg1, E12 means X.Arg1=Y.Arg2 and so on.

3.2.2 Co-occurrence Calculation

When we consider the co-occurrence both in one document and over multiple documents, we define the quality of co-occurrence as

$$C(x, y) = \frac{1}{2} [f(x, y) / f(x) + f(x, y) / f(y) + g(x, y) / [f(x) + f(y)]] \quad (1)$$

where $f(x,y)$ is a count for two tuples occurring in the same document, $f(x)$ is the total count that x occurs in each document; $f(y)$ is the total count that tuple y occurs in document; $g(x,y)$ is the co-occurrence count over the documents for x and y which is normalized due to the permutation and combination. We consider the distance which may influence the co-occurrence quality. Suppose (x,y) is a pair of tuples that occur in one document, the co-occurrence distance factor is defined as

$$d(x, y) = e^{-\gamma(k-1)} \quad (2)$$

where γ is the revision parameter, usually take the value of 0.5. The co-occurrence value is thus represented as

$$C'(x, y) = C(x, y) \cdot d(x, y) \quad (3)$$

3.3 Co-occurrence Graph Analysis

We construct a co-occurrence graph $G=(V,E)$ whose nodes are relation tuples and edges are weighted co-occurrence value $C'(x,y)$. Figure.3 shows a co-occurrence graph. If $C'(x, y) \geq \lambda$, where λ is the threshold, then link the x node with node y .

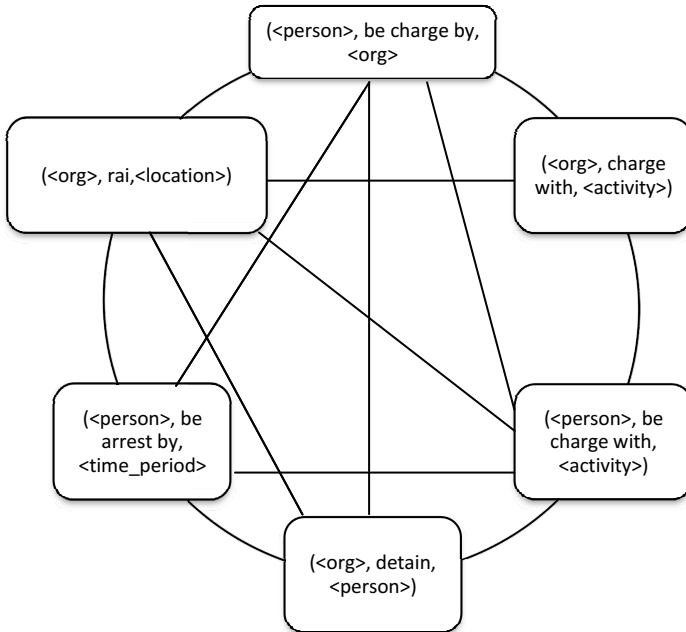


Fig. 3. Part of a graph showing tuples strongly associated with (<person>, be charge by, <org>)

For each cluster of co-occurrence graph, we perform graph analysis to find the most related nodes and use them to create an event schema. Page rank algorithm can be well adapted to our graph analysis. We use the Stanford Personalized PageRank algorithm¹ to rank the graph and obtained the top n nodes from the graph which may establish an event schema. For the top ranking node tuple X: (Arg1, *Rel*, Arg2), we record two roles (R1, R2) as the roles of the schema and use *Rel* to be the relation they have in the event. Then, we merge similar roles into one role according to Niranjana and Oren(2013).

Finally, the event schema generation is done with a ranking list of tuples. And we take the top n elements to be represented as part of the event schema.

4 Experiments

In this experiment we are desired to know whether the schema generated by our method has coherence of real world, such as common topic and correct roles. Since the output is large due to the large-scale corpus, we sample some the output schemas and examine whether they make sense in the real world.

4.1 Evaluation Methods

Is there an underlying common topic or event among the elements of the event schema? We focus on two measures, *topic coherence* and *role coherence* as mentioned in Niranjana and Oren(2013). A good schema must be topically coherent, i.e., the relations and roles should relate to some realworld topic or event. The tuples that comprise aschema should be valid assertions that make sense in the real world. Finally, each role in the schema should belong to a cohesive set that plays a consistent role in the relations. We compare event schemas considered relational tuples co-occurrence over multiple documents against schemas released byNiranjana².

Topic coherence is aimed to test whether the relations in a schema form a coherent topic or event, we presented the annotators with a schema as a set of grounded tuples, showing each relation in the schema, but randomly selecting one of the top n instances from each role. We collected five instantiations like this for every schema. Three questions are ready for annotators: (1) is each of the grounded tuples meaningful in the real world; (2) do the majority of relations form a coherent topic; and (3) does each tuple belong to the common topic.

Role Coherence is aimed to test whether the instances of a role form a coherent set. We held the relation and one role fixed and presented the annotators with the top 5 instances for the other role. For each event schema's element (Arg1, Relation, Arg2), for example, we fixed the Relation and Arg1 and tested the top n instances of Arg2 to ask the annotators whether it belongs to the topic.

¹ Available on <http://nlp.stanford.edu/projects/pagerank.shtml>

² Available at <http://relgrams.cs.washington.edu>

We are also interesting to compare our event schemas with the MUC-4 templates. One of the MUC-4 tasks is to extract information about terrorist events, such as the names of perpetrators, victims, instruments, etc. MUC-4 templates for terrorist events include bombing, attack, kidnapping, and arson. Each template has six slots: perpetrator, victim, physical target (omitted for kidnapping), instrument (omitted for kidnapping and arson), location and date. We picked out the event schemas that include the key words of the tuple: (bomb, explode at, <location>) (bomb, explode on, <time_unit>) (<person>, plant, bomb) (bomb, wound/kill, <person>) to describe the bombing template. We checked the roles of the schema as the comparison to MUC-4 template slots.

4.2 Results

We found that 96% of the schemas have sense in the real world, and 94% of the roles are reasonable and not mixed up in one schema. Two evaluation tasks are performed to test the coherence and validity of the event schema and the roles.

We obtained a test set of 10000 schemas per system by randomly sampling from each system. We evaluated this test set by manual testing whether the relations in one schema show a coherent topic or event. We took an average of ratings from five annotators as the final annotation. Figure 4 and Figure 5 show us the topic coherence and role coherence respectively.

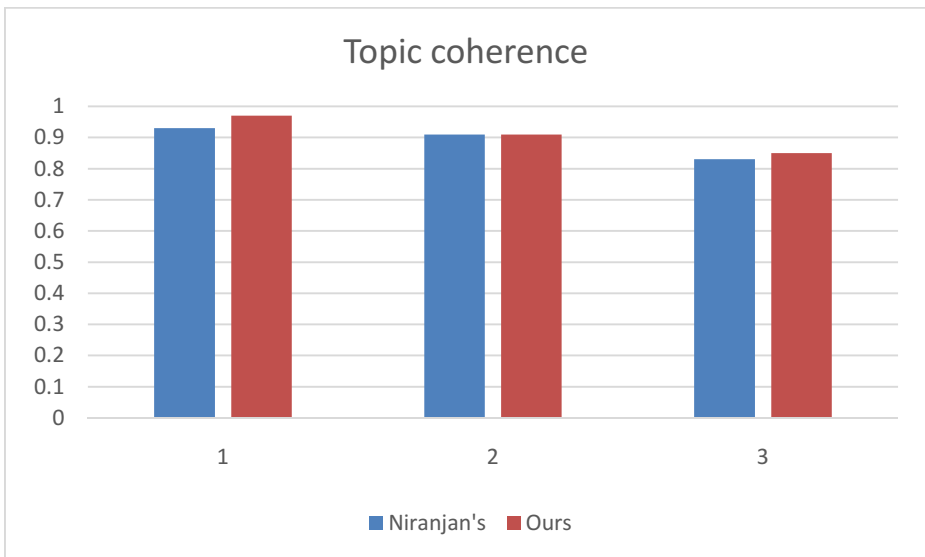


Fig. 4. Topic coherence bar gram shows whether the topic makes sense in the real-world. Histogram 1: percentage of schema instantiations with a coherent topic; Histogram 2: percentage of grounded valid tuple that assert valid real-world relations; Histogram 3: percentage of grounded statements where the instantiation has a coherent topic and the tuple is valid.

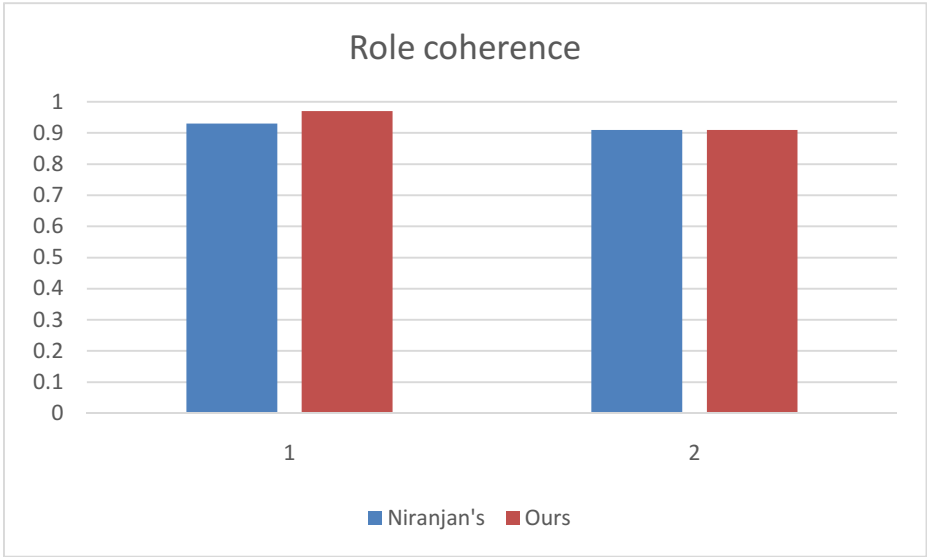


Fig. 5. Role coherence bar gram shows the roles in event schema correspond to common sense or not. Histogram 1: the percentage of tested roles of event schema which are coherent. Histogram 2: the percentage of top role instances which make sense in real-world.

Topic coherence is shown by topic coherence bars in Figure 3. Our result has a little advantage compared to Niranjan’s. Relational tuples co-occurrence over multiple documents can add the weight between nodes and graph analysis can find the most relational nodes correctly. As a result, some event schemas may be richer and more coherent in our real world. Role coherence bars in Figure 4 shows the instances of roles that corresponding to the slots of the schema. Co-occurrence may link some roles over multiple documents and the role candidates are more complete to be induced.

The result compared with MUC-4 templates is shown in Table 1. A proportion of slots correctly discovered for each MUC-4 terrorist event template is represented. We can see that bombing event reflect the correspondence with MUC-4 template slots, as six slots are available at high proportion. We examined the corpus and the co-occurrence graph and found that relational tuples co-occurrence over multiple documents was helpful and assisted the graph analysis to extract and merge the bombing event element. Kidnapping event is weaker to discover the location slot as the co-occurrence both in one document and over multiple documents is not that tight.

Table 3. A proportion of slots discovered for each MUC-4 terrorist event template

Template	perpetra- tor	victim	Physical target	instru- ment	location	date
bombing	0.935	0.923	0.912	0.931	0.957	0.953
attack	0.928	0.939	0.914	0.945	0.967	0.962
kidnapping	0.892	0.921	N.A.	N.A.	0.630	0.840
arson	0.921	0.933	0.870	N.A.	0.920	0.933

5 Conclusion

We exploit the relational tuples co-occurrence both in one document and over multiple relational documents and build a relational graph with the nodes of the form (Arg1, relation, Arg2). Relational tuples co-occurrence over multiple documents may help to simplify the event schema as pointed in Section 1. Besides, relational tuples co-occurrence over multiple documents can find more instances for roles in event schema which may enrich the event and make it more coherent in the real world. We would like to investigate event schema induction evaluation, for example to evaluate event coherence automatically in addition to the slots and entities.

Acknowledgement. This paper is supported by National Key Basic Research Program of China 2014CB340504 and NSFC project 61375074.

References

1. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative event chains. In: Proceedings of ACL 2008: HLT (2008)
2. Chambers, N., Jurafsky, D.: Unsupervised learning of narrative schemas and their participants. In: Proceedings of ACL (2009)
3. Chambers, N., Jurafsky, D.: Template-based information extraction without the templates. In: Proceedings of ACL (2011)
4. Balasubramanian, N., Soderland, S., Mausam, Etzioni, O.: Generating Coherent Event Schemas at Scale. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
5. Balasubramanian, N., Soderland, S., Mausam, Etzioni, O.: Rel-grams: A Probabilistic Model of Relations in Text. In: Proc. of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX) (2012)
6. Cheung, J., Poon, H., Vandervende, L.: Probabilistic frame induction. In: Proceedings of NAACL HLT (2013)
7. Barzilay, R.R.: Multi Event Extraction Guided by Global Constraints. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2012)

Negation and Speculation Target Identification

Bowei Zou, Guodong Zhou, and Qiaoming Zhu

Natural Language Processing Lab, School of Computer Science and Technology
Soochow University, Suzhou, 215006, China

zoubowei@gmail.com, {gdzhou, qmzhu}@suda.edu.cn

Abstract. Negation and speculation are common in natural language text. Many applications, such as biomedical text mining and clinical information extraction, seek to distinguish positive/factual objects from negative/speculative ones (i.e., to determine what is negated or speculated) in biomedical texts. This paper proposes a novel task, called negation and speculation target identification, to identify the target of a negative or speculative expression. For this purpose, a new layer of the target information is incorporated over the BioScope corpus and a machine learning algorithm is proposed to automatically identify this new information. Evaluation justifies the effectiveness of our proposed approach on negation and speculation target identification in biomedical texts.

Keywords: negation, speculation, target identification.

1 Introduction

Negative and speculative expressions are common in natural language text. While negation is a grammatical category which comprises various kinds of devices to reverse the truth value of a proposition, speculation is a grammatical category which expresses the attitude of a speaker towards a statement in terms of degree of certainty, reliability, subjectivity, sources of information, and perspective. It is widely accepted that negation and speculation play a critical role in natural language understanding, especially information extraction from biomedical texts. Szarvas (2008) observes that a significant proportion of the gene names mentioned in a corpus of biomedical articles appear in speculative sentences (638 occurrences out of a total of 1,968). Morante and Sporleder (2012) state that in order to automatically extract reliable information from clinical reports, it is of great importance to determine whether symptoms, signs, treatments, outcomes, or any other clinical relevant factors are present or not.

Recent studies of negation and speculation on biomedical information extraction focus on trigger detection, which aims to detect the signal of a negative or speculative expression, and scope resolution, which aims to determine the linguistic coverage of a negative or speculative trigger in a sentence, in distinguishing unreliable or uncertain information from facts. For example, sentences (1) and (2) include a negative trigger and a speculative trigger, respectively, both denoted in **boldface** with their scopes denoted in square brackets (adopted hereinafter).

- (1) *Our results show that [no transcription of the RAG-1 gene could be detected].*
 (2) [The cardiovascular disease *may recur*] even after cure.

However, people may wonder what is exactly negated or speculated, e.g. which objects are negated or speculated on a clinical medicine, by which for an authority to make a proper action. This poses strong requirements beyond trigger detection and scope resolution.

From this regard, we propose a novel task, called negation and speculation target identification, to extract the object targeted by a negative or speculative expression. For example, in sentences (1) and (2), the targets are *transcription of the RAG-1 gene* and *the cardiovascular disease*, given the negative and speculative expressions **no** and **may**, respectively (denoted with underline). The main contributions of this paper are the proposal of a new task, the annotation of a new corpus and the proposal of a machine learning approach for such a new task. It is also worthy to mention that negation and speculation target identification can not only complement trigger detection and scope resolution but also help them better infer unreliable or uncertain information from context. For example, in the scenario of sentence (1), if sentence *Some studies claimed that transcription of the RAG-1 gene was detected* is given as its context, we can easily infer that the affirmative statement in the given context is doubtful.

The rest of this paper is organized as follows. Section 2 introduces the related work. In Section 3, we discuss some details on negation and speculation target identification. In Section 4, the annotation guidelines for the target identification corpus are introduced. In Section 5, our machine learning approach is proposed with various kinds of lexical and syntactic features. Section 6 reports the experimental results and gives some discussions. Finally, we draw the conclusion in Section 7.

2 Related Work

There is a certain amount of literature within the natural language processing community on negation and speculation. While earlier studies adopt rule-based approaches (e.g., Light et al., 2004), machine learning-based approaches begin to dominate the research on negation and speculation (e.g., Morante et al., 2008) since the release of the BioScope corpus (Vincze et al., 2008).

Recently, the studies on negation and speculation have been drawing more and more attention, such as the CoNLL'2010 Shared Task on trigger detection and scope resolution of negation and speculation (Farkas et al., 2010), and the ACL 2010 Workshop on negation recognition (Morante and Sporleder, 2010). Even more, a special issue of Computational Linguistics (Morante and Sporleder, 2012) has been published on negation and speculation. However, none of the above shared tasks or workshops aim at identifying the target of a negative or speculative expression.

Similar to target identification in biomedical texts, opinion target extraction (OTE) in sentiment analysis aims to identify the topics on which an opinion is expressed (Pang and Lee, 2008). Nevertheless, above opinion target is related to a sentiment word instead of a negative or speculative expression. Among others, in semantic role labeling (Carreras and Màrquez, 2005), a target may act as some semantic role. How-

ever, such correspondence does not always exist since a target is dominated by a negative or speculative expression, while a semantic role is dominated by a predicate.

Even though the studies on negation and speculation have received much interest in the past few years, open access annotated resources are rare, usually with limitation in information and small scale in size. For example, the Hedge Classification corpus (Medlock and Briscoe, 2007) only contains the annotation for hedge triggers in 1537 sentences and does not contain the scope information. The BioScope corpus (Vincze et al., 2008) annotates the linguistic scopes of negative and speculative triggers in biomedical texts. Obviously, none of above resources is suitable for negation and speculation target identification.

3 Target of Negation and Speculation

A negative and speculative expression always attaches to an object or its attribute which is negated or speculated. In this paper, we define such an object as the target of negation or speculation (except particular illustration, we use “target” for simplicity).

According to above definition of target, it seems that almost all targets should be entities. However, the fact is that some predicates can be also negated or speculated by a verbal negation or speculation expression. For example, in sentence (2), the cardiovascular disease is the speculation target, meaning whether this disease could recur. Here, the direct speculative object is an event (recur). In such a situation, we consider the agent of a negative or speculative expression as the target. Statistics on 100 samples randomly chosen from our target identification corpus (For details please refer to Section 4) shows that only 42 targets are entities, while the remaining 58 targets are the agents of verbal negative or speculative expressions. To better illustrate the concept of target, we clarify its difference with scope and subject.

Target vs. Scope: Both scope and target are extremely important to capture the negative and speculative meanings. While scope refers to the grammatical part in a sentence that is negated or speculated, target is concerned with the negative or speculative object rather than the grammatical coverage of a negative or speculative cue. For example, in sentence (1), while the scope is *no transcription of the RAG-1 gene could be detected*, representing a negative proposition, the target is *transcription of the RAG-1 gene*, representing a negated object. In addition, it should be noted that a target is not always in a scope. For example, in sentence (3), while the scope *without voting* negates the evaluating way for prize, *The Prize of Best Employee* is target.

(3) *The Prize of Best Employee* is awarded [*without voting*], unexpectedly.

(4) Company management has **not** yet decided on *the Prize of Best Employee*.

Target vs. Subject: The target and the subject in a sentence may not be the same. A target represents the object described by a negative or speculative expression, while a subject is a constituent that conflates nominative case with the topic. The former is from semantic perspective on negation and speculation, while the latter is from the syntactic perspective. In sentences (3) and (4), both the negated targets are *the Prize of Best Employee*, no matter whether or not they are the subjects in a sentence.

4 Corpus

Due to the lack of corpus annotation for target identification on negation and speculation, a new layer of the target information is added to the BioScope corpus (Vincze et al., 2008)¹, a freely available resource which has already been annotated with the linguistic scopes of negative and speculative triggers in biomedical texts.

4.1 Annotation Guidelines

In BioScope corpus, only the sentences including the speculative or negative information are chosen for target annotation. In annotation, the most general two basic guidelines are: 1) if a noun phrase can be inferred as the object described by a negative or speculative expression, it is the target. 2) Otherwise, target is the agent of the sentence concluding negative or speculative trigger. During the process of annotation, more than 70% of sentences can be annotated by the two basic guidelines.

In the following, we introduce the specific guidelines developed throughout the annotation process with examples to deal with the specific characteristics in target identification on negation and speculation.

Guideline 1: In sentence (5), the target should be something (maybe drug or therapy), but does not appear in the sentence. In this situation, we annotated *it* as the target, since this paper is only concerned with the target in a sentence, and as for what it is actually, there is no need for annotation.

(5) *It is **not** effective for all tuberculosis patients.*

Guideline 2: When there is a raising verb (e.g., seem, appear, be expected, be likely, etc.) in a sentence, as in sentence (6), we prefer to mark the logical agent as the target rather than the formal one.

(6) *It **seems** that the treatment is successful.*

Guideline 3: A target can be partly determined on the basis of syntax. Our manual statistics on syntactic category shows that noun phrases exist in 97.59% and 98.14% of the targets on negation and speculation respectively. Besides, in the annotation process, we extend their scopes to the biggest syntactic unit as much as possible due to following two facts:

First, taking into account the information integrality of a target, it seems better to include all the elements attached to the target, such as prepositional phrases, determiners, adjectives, and so on. In sentence (7), with *blood lymphocytes* as the head word of the target, the two prepositional phrases (introduced by *from* and *with*) that represent the target's attributes are also included within the target:

(7) *In contrast, blood lymphocytes from patients with granulomatous diseases have **little** effect on children.*

Second, the status of a modifier is sometimes uncertain. For example, the negative trigger **no** in sentence (8) could modify two different semantic elements: On one

¹ <http://www.inf.u-szeged.hu/rgai/bioscope>

hand, it may modify *primary*, with the meaning *the glucocorticoid metabolism is impaired*. On the other hand, it may modify *impairment*, with the meaning *there is no impairment of the glucocorticoid metabolism at all*. We cannot resolve such ambiguity on the basis of contextual information. Fortunately, we can avoid such ambiguity with the maximal length annotation strategy. Furthermore, if the category of target could be directed further fine by a modifier, the target should contain the modifier.

(8) *There is **no** primary impairment of glucocorticoid metabolism in the asthmatics.*

Guideline 4: When the trigger is a conjunction, we extend the target all members of the coordination.

(9) *In common sense, symptoms include fever, cough or itches.*

Guideline 5: If a target contains an omitted part, for simplicity, we avoid completing it.

(10) *Finally, recombinant GHF-1 interacted directly with c-Jun proteins but **not** c-Fos.*

Guideline 6: If there are punctuation marks or conjunctions at the head or end of a target, we ignore them. Nevertheless, for coherence, the punctuation marks or conjunctions in the middle of the target are included (see sentence (9)).

4.2 Corpus Annotation

We have annotated 1,668 negative sentences and 2,678 speculative sentences by two independent annotators following the guidelines in Section 4.1 over the BioScope corpus. Table 1 summarizes the chief characteristics of the corpus.

Table 1. Statistics of target identification corpus

		Negation	Speculation
#Sentence		1668	2678
%In scope		54.98%	63.71%
%Out of scope		45.02%	36.29%
Average length of sentences		29.73	31.16
Average length of targets		4.36	5.27
Relation to target and keyword	%Before	54.80%	43.05%
	%After	45.02%	49.48%
%Noun phrase target		98.02%	97.46%

During the annotation process, annotators can only refer to the negative or speculative triggers but not their corresponding scopes. This is necessary to ensure that the annotation is not biased by scope information provided of BioScope corpus. The 3rd and 4th rows in Table 1 show the ratio of the cases whether the targets are involved in the scope. It indicates that the targets are not always in the scope.

In target identification corpus, if there is more than one trigger in a sentence, we treat them as different instances. The 7th and 8th rows show the ratio of the cases whether the target position is in front of the trigger or behind it. Such close ratios show that the positional relation between the trigger and its target is not apparent.

Additionally, according to our statistics, there are 97.59% and 98.14% of the targets including a noun phrase on negation and speculation respectively (the 9th row in Table 1). This is the reason that we regard the noun phrases as target candidates in our experiments.

The annotators are not allowed to communicate with each other until the annotation process is finished, but they could appeal to linguists when needed. Differences between the two annotated results are also resolved by linguists.

A checking step can ensure that the annotation is grammatical. In this step, every instance has been processed with a syntactic parser (refer to Section 6.1). If the maximal syntactic parsing sub-tree of all target terminal nodes has other terminal nodes, the annotation system would require annotator to confirm. For example in Figure 1, the least common ancestor of “*cotransfection, TCF-1*” is the syntactic category node “S”, but “S” includes other terminal nodes (e.g., “*Upon*”). This kind of instance would be re-annotated. About 17% of negative instances and 12% of speculative instances are re-annotated.

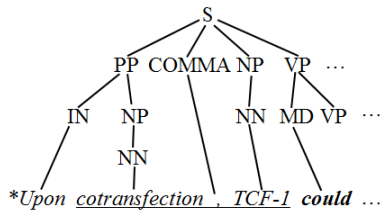


Fig. 1. The instance that needs to re-annotate

We measured the consistency level of the annotation by inter-annotator agreement analysis. The Cohen’s kappa statistic (Conger, 1980) for our annotation is 0.83. After the careful examination of the disagreements by linguists, they are resolved. The main conflict is whether the modifiers of the target (mentioned in Section 4.1) are involved.

5 Methodology

We propose a machine learning approach with various kinds of lexical and syntactic features for negation and speculation target identification.

Features. By taking the noun phrase in sentence as target candidate, we adopt ranking Support Vector Machine (rSVM) model for training. To capture more useful information, we propose various kinds of refined features from lexical and syntactic perspective. Table 2 lists these features.

Features (1-5) are the basic information of negative/speculative trigger and target candidate, including trigger itself and its part-of-speech (POS), candidate itself, the head word of candidate and its POS. For feature S1 and S2, if the trigger or its POS are different in two instances, the corresponding targets are likely to act different roles in sentence. Considering the triggers in sentence (11) and (12), for trigger *hypothesis* and its POS *NN*, target is more likely present in predicative; for trigger *assumes* and

its POS *VBZ*, target is likely to be the subject of its clause. And likewise, the head word features are also informative for target.

Features (6-9) are the syntactic relationship between target candidate and negative/speculative trigger, including syntactic path, relative position, distance in parsing tree, and distance of tokens. For feature S6, the syntactic path feature provides structural information of parsing tree, but it is sparse when candidate is far away from the trigger. Although feature S7 is simple, it relates the voice of a verbal trigger. For example, considering the verbal speculative triggers *be suggesting* and *be suggested*, the role of target is likely to be different.

Table 2. Full set of lexical and syntactic features on target identification

No.	Feature	Explanation
S1	keyword	keyword itself
S2	keyword_POS	keyword's part-of-speech
S3	candidate	candidate itself
S4	headword	candidate's head word
S5	headword_POS	part-of-speech of candidate's head word
S6	syn_path_keyword	syntactic path between keyword and candidate
S7	PR_keyword	positional relationship of candidate with keyword
S8	syn_dis_keyword	syntactic distance from candidate to keyword
S9	word_dis_keyword	word distance from candidate to keyword
S10	left_phrase_type_tag	left sibling tag of candidate's syntactic category
S11	right_phrase_type_tag	right sibling tag of candidate's syntactic category
S12	left_phrase_type_seq	sequence of words govern by left sibling of candidate's syntactic category
S13	right_phrase_type_seq	sequence of words govern by right sibling of candidate's syntactic category
S14	nearest_verb	nearest verb with keyword in syntactic parsing tree
S15	PR_SF11	positional relationship of candidate with SF11 verb
S16	syn_dis_SF11	syntactic distance from candidate to SF11 verb
S17	word_dis_SF11	word distance from candidate to SF11 verb
S18	syn_path_SF11	syntactic path between SF11 verb and candidate
C1	S1 + S14	
C2	S7 + S15	
C3	S8 + S16	
C4	S9 + S17	

Features (10-13) are the adjacent syntactic features of target candidate, including the left and right syntactic categories, and the left and right chunks. Intuitively, these features are sparse and featureless on lexical level (S12 and S13), but not on syntactic level (S10 and S11).

Feature (14-18) are the syntactic information associated with the verb, which may have the directly relatedness between negative/speculative trigger and its corresponding target. Motivating in part by semantic role labeling (SRL), we infer that features related to verb in sentences are effective for target identification. That is because, in SRL, the predicate verb involves lots of dominated and modified relationship between itself and other semantic roles. Similarly, the negative or speculative triggers are

closely connected with the verb on syntactic structures. For this reason, we explore the features from the verb for getting more supplementary syntactic information.

Since above features may not work on target identification of both negation and speculation with equal effectiveness, we adopt a greedy feature selection algorithm as described in Jiang et al. (2006) to pick up positive features incrementally according to their contributions on the performance of our system. The algorithm repeatedly selects one feature each time, which contributes most, and stops when adding any of the remaining features fails to improve the performance.

Post-Processing. As mentioned in Section 4.1, a target is the object described by a negative or speculative trigger. According to the annotation guidelines, we adopt the maximal principle to label targets, which involve some modified structures, such as prepositional phrase and attributive clause. Our classification system takes noun phrases as instances, but in fact, some syntactic structures of targets are NP+PP but not NP. In that case, we cannot get correct results. For this reason, we propose a post-processing step to improve performance, described in Algorithm 1 below. In post-processing, if the syntactic category of prediction has a right sibling of PP or SBAR, we connect the sibling to the prediction and continue to check the rest.

Input:
 syntactic parsing tree: T ,
 prediction node: N_{pred}

Output:
 word sequence of N_{pred} : W_{pred}

Initialize:
 $W_{pred} = \text{get_sequence}(N_{pred});$
 $N_{sibling} \leftarrow \text{get_right_sibling}(N_{pred});$

while $N_{sibling} \neq \text{NULL}$
 ⋮
 ⋮
if $N_{sibling} = \text{"PP"}$ or "SBAR" **then**
 $W_{pred} \leftarrow W_{pred} + \text{get_sequence}(N_{sibling});$
end
 $N_{sibling} \leftarrow \text{get_right_sibling}(N_{sibling});$
 ⋮
 ⋮
End

Algorithm 1. Post-processing algorithm

6 Experimentation

6.1 Experimental Settings

Dataset: In consideration of the features selection, we have split the corpus into 5 equal parts, within which 2 parts are used for feature selection and the rest for experiments. On the one hand, in feature selection, we divide the data into 5 equal parts, within which 4 parts for training and the rest for developing. We divide the experimental data into 10 folds randomly, so as to perform 10-fold cross validation.

Syntactic Parser: All sentences in our corpus are tokenized and parsed using the Berkeley Parser (Petrov et al, 2007)² which have been trained on the GENIA Tree-Bank 1.0 (abbr., GTB; Tateisi et al., 2005)³, a bracketed corpus with PTB style in biomedical field. 10-fold cross-validation on GTB1.0 shows that the syntactic parser achieves 87.12% in F1-measure.

Classifier: We selected the SVM^{Light}⁴ with the default parameters configuration as our classifier.

Evaluation Metrics: Exact match is used to evaluate the correctness of a target (Accuracy, abbr., Acc). That is to say, an extracted target is considered as correct only if it has exactly the same span boundaries as the annotated ones in gold standard. Additionally, we adopt Precision (P), Recall (R), and F1-measure (F1) as evaluation metrics. The accuracy which takes sentence as a unit measures the performance of our system. The PRF-measure which takes target candidate as a unit reports the performance of the binary classifier by which every instance has been classified.

6.2 Results of Target Identification

Performance of Baselines. We implemented four baselines to measure the difficulty of the target identification task:

- **Baseline_First:** select the first noun phrase in sentence as target.
- **Baseline_Last:** select the last noun phrase in sentence as target.
- **Baseline_Longest:** select the longest noun phrase in sentence as target.
- **Baseline_Nearest:** select the noun phrase which is nearest to the trigger as target. The distance is measured by syntactic path. For example, in $NP > S < VP < VBN$, the distance from NP to VBN is 3.

Table 3 lists performances of baseline systems without post-processing. It shows that the performances of *Baseline_Longest* and *Baseline_Nearest* are higher than the other two systems. The two former baselines do not consider the relationship between trigger and target, which is direct clue for target identification. However, the *Baseline_Longest* system adopts no information involving trigger either, but its performance improves a little. We infer that the longest noun phrase in a sentence involves many modifiers which are always the object most impacted by the trigger. For both negation and speculation target identification, the *Baseline_Nearest* system achieves the best performance. It indicates that the syntactic path characteristics are effective to detect the target dominated by trigger. Inspired by the *Baseline_Nearest* system, we employ some syntactic path features in our classification.

Table 4 shows the effectiveness of our post-processing algorithm described in Section 5.3. All of baselines greatly improve by the post-processing algorithm. Besides, it is worth noting that the *Baseline_Longest* system only improves of less than 2 and 3 in accuracy on negation and speculation, respectively, largely due to the completeness of the longest noun phrase in a sentence.

² <http://code.google.com/p/berkeleyparser>

³ <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA>

⁴ <http://svmlight.joachims.org>

Table 3. Accuracy of baselines

System	Neg	Spe
<i>Baseline_First</i>	17.68	23.37
<i>Baseline_Last</i>	21.04	18.29
<i>Baseline_Longest</i>	30.70	32.64
<i>Baseline_Nearest</i>	31.95	36.63

Table 4. Accuracy of post-processing

System	Neg	Spe
<i>Baseline_First</i>	21.44	25.37
<i>Baseline_Last</i>	27.41	24.19
<i>Baseline_Longest</i>	32.13	35.59
<i>Baseline_Nearest</i>	39.93	43.77

Our Performance. We perform a greedy algorithm as described in Section 5 to select a set of effective syntactic features on Feature Selection dataset. Table 5 and 6 show the effects of selected features in an incremental way for negation and speculation respectively. We also employ all of the features described in Table 2 for target classification.

Table 5. Performance improvement of features incrementally on negation

Features	P	R	F	Acc
S6	45.29	33.64	38.61	40.31
+ S3	51.77	46.98	49.26	50.02
+ C2	60.40	51.19	55.41	57.06
+ S11	66.58	56.79	61.30	61.21
+ S8	71.18	58.44	64.18	63.92
+ C1	73.51	60.08	66.12	64.39
ALL	68.01	55.37	61.04	59.23

Table 5 shows that the system with feature set of {S6, S3, C2, S11, S8, and C1} achieves the best performance. Table 6 shows that the system the feature set of {S6, S3, S11, S9, S8, S15, and S7} achieves the best performance.

Table 6. Performance improvement of features incrementally on speculation

Features	P	R	F	Acc
S6	59.38	48.86	53.16	54.19
+ S3	68.41	55.91	61.53	64.23
+ S11	73.28	59.50	65.67	67.36
+ S9	75.42	62.64	68.42	68.84
+ S8	76.63	62.98	69.14	69.09
+ S15	77.21	63.26	69.54	69.35
+ S7	78.03	63.55	70.05	69.37
ALL	76.16	59.48	66.79	67.99

It is worth noting that the features S6, S3, S11 and S8 are effective both on negation and speculation. Feature S6 directly represents the connecting pathway between trigger and target. For instance, on negation, corresponding a preposition trigger (e.g., “without”), the path is usually “PP<NP”. On speculation, if the trigger is a modal verb (e.g., “might”), the syntactic path between trigger and target probably is “NP>S<VP” or “VP<NP”. Feature S3 is the target candidate itself. In the same topic or discourse of a literature, the depicted target is likely to be concentrated. If a target

is negated or speculated, the one in other sentences may also have the same semantic representation (negation or speculation).

Additionally, features S11 and S8 also have a little effect for target classification on both negation and speculation. In our corpus, 11.8% of negative triggers of the total are “no (det)” and 4.7% of negative triggers are “without”. This kind of triggers usually takes a right sibling as their targets. Thus, feature S11 can dig the characteristics in this situation. Similar to feature S6, feature S8 also represents the syntactic relatedness between trigger and target.

Features C2 and C1 are the particular features on negation target identification. They are related to trigger and its nearest verb in syntactic parsing tree. It shows that the position of target suffers from the combined impact of trigger and its corresponding verb.

Features S9, S15, and S7 are the particular features on speculation target identification. Similar to feature S8, feature S9 is another kind of distance between trigger and target. Features S15 and S7 are the target candidate’s position to verb and trigger respectively.

Table 7. Performance of target identification system on negation and speculation

	P	R	F	Acc
Negation	76.27	63.53	69.32	70.13
Speculation	84.32	69.85	76.41	74.46

Table 7 shows the performance of our target identification system with post-processing. It significantly improves the accuracy by 5.74 from 64.39 to 70.13 on negation ($p < 0.05$) and by 5.09 from 69.37 to 74.46 on speculation ($p < 0.05$). It indicates that not all targets are noun phrases and the post-processing algorithm we proposed is instrumental.

7 Conclusion

In this paper, we propose target identification on negation and speculation, a novel task on negation and speculation in biomedical texts. Due to the lack of corpus, we add a new layer of the target information over the BioScope corpus. On the basis, a set of features are depicted and a supervised model is proposed to implement target identification on negation and speculation. The experimental results show that syntactic features play a critical role in capturing the domination relationship between a negative or speculative trigger and its target.

In future work, we will finalize and release the corpus and explore more useful features for target identification on negation and speculation. Moreover, we will systematically explore its application in other domains, e.g., legal or socio-political genre.

Acknowledgments. This research is supported by the National Natural Science Foundation of China, No.61272260, No.61331011, No.61273320, the Natural Science Foundation of Jiangsu Province, No. BK2011282, and the Major Project of College

Natural Science Foundation of Jiangsu Province, No.11KIJ520003. The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

1. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In: Proceedings of the 9th Conference on Computational Natural Language Learning, pp. 152–164 (2005)
2. Conger, A.J.: Integration and generalization of kappas for multiple raters. *Psychological Bulletin* 88(2), 322–328 (1980)
3. Farkas, R., Vincze, V., Móra, G., Csirik, J., Szarvas, G.: The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Uppsala, Sweden, pp. 1–12 (2010)
4. Jiang, Z., Ng, H.T.: Semantic Role Labeling of NomBank: A Maximum Entropy Approach. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 138–145 (2006)
5. Light, M., Qiu, X., Srinivasan, P.: The Language of Bioscience: Facts, Peculations, and Statements in Between. In: Proceedings of the HLT BioLINK 2004, pp. 17–24 (2004)
6. Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, pp. 992–999 (2007)
7. Morante, R., Liekens, A., Daelemans, W.: Learning the scope of negation in biomedical texts. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, pp. 715–724 (2008)
8. Morante, R., Sporleder, C. (eds.): Proceedings of the Workshop on Negation and Speculation in Natural Language Processing. University of Antwerp, Uppsala (2010)
9. Morante, R., Sporleder, C.: Modality and Negation: An Introduction to the Special Issue. *Computational Linguistics* 38(2), 223–260 (2012)
10. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis: Foundations and Trends. *Information Retrieval* 2(12), 1–135 (2008)
11. Petrov, S., Klein, D.: Improved Inference for Unlexicalized Parsing. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, pp. 404–411 (2007)
12. Szarvas, G.: Hedge classification in biomedical texts with a weakly supervised selection of keywords. In: Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics, pp. 281–289 (2008)
13. Tateisi, Y., Yakushiji, A., Ohta, T., Tsujii, J.: Syntax Annotation for the GENIA Corpus. In: Proceedings of IJCNLP, Companion Volume, pp. 222–227 (2005)
14. Vincze, V., Szarvas, G., Farkas, R., Mora, G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics* 9(suppl. 11), S9+ (2008)

An Adjective-Based Embodied Knowledge Net

Chang Su*, Jia Tian, and Yijiang Chen

Department of Cognitive Science,
Xiamen University, Xiamen 361005, China
suchang@xmu.edu.cn

Abstract. As the findings about the embodiment of language comprehension and some difficulties in the existing models of metaphor processing, this paper presents an adjective-based embodied cognitive net, which constructs the comprehension of knowledge in a novel view. Different from the traditional way that takes concepts as the core of knowledge comprehension, this paper views the emotions as the core and the motive power that human beings knowing the world. It is claimed that the adjective is the carrier of emotion in this paper, rather the concept. From the very nature, while getting a new thing, the first thing that comes to human's mind are the original descriptions(usually are adjectives) and then are the concepts. Thus, this paper constructs a net based on adjectives from concrete to abstract according to the embodiment. In this knowledge net, nouns are contained as the attachment to construct a mapping between adjectives and concepts. Specially, this paper gives the embodied emotion to the adjective to deal with the emotion inference and metaphor emotion analysis in the future work.

Keywords: adjective, emotion, embodiment, knowledge net.

1 Introduction

1.1 Embodiment

Recent researchers in cognitive science claim that knowledge obtaining and comprehension are embodied. [1] presented "embodied philosophy" and "embodied mind" and gave three hypothesis about embodiment: 1) Mind is embodied; 2) Thought is unconscious; 3) The abstract concepts are mostly metaphorical. Lakoff emphasized the foundational effect of mind and cognition and claimed that the body, brain and environment are the cognitive basis of human's inference. This is the *embodied realism*, which holds the view of that the human's body is under a interaction related to their concept system. Based on the *embodied realism*, [2] pointed the effect of body's sight and action system to the cognitive language. [3] claimed that it is necessary to know the world with human's body. Since then, embodiment has been applied into language comprehension. [4] presented the embodied semantics and pointed that the language containing

* Corresponding author.

phonology and grammar is embodied. [2] discussed the embodiment of syntax in the view of the real world events. [5] considered that the specific events are converted to some abstract stories with the help of sensory modalities and body action, which are the basis of syntactic structure.

1.2 Emotion

Emotion is a controversial topic in psychology research. It is a basis of forceful conversation and divergence according to the earliest philosophers up to the present day [6]. As the development of NLP, the emotion of textual data is becoming more and more important and how to classify emotions in a large-scale text is becoming one of the topics in text analysis. [7] gave an emotion recognizing method by knowledge based artificial neural network. [8] gave an approach in estimating textual emotion using keywords based on searching. As the importance of metaphor in NLP and the findings in cognitive science, researchers realize that the metaphor is emotional and it is essential to contain emotion analysis in metaphor processing. Emotions in metaphorical expressions also reflect the cognitive nature of metaphors.

This paper constructs a novel adjective-based knowledge comprehension net in a view of embodiment. Different from previous work, this net focuses on the embodiment of adjectives. We consider that during the process of human beings knowing the world from childhood and the improvement of human society, adjectives gather human's cognition to the whole world. Given a new thing, human beings firstly come up with the descriptions of it, which are usually adjectives instead of nouns. Thus, we claim that adjectives are more embodied than traditional points of concepts (nouns).

As human's cognition to the world is emotional, we consider that the adjectives are apparently emotional. Different from the traditional way that divides emotions into six primitive types (i.e. joy, love, surprise, anger, sadness and fear [9] or simply into negative, positive and neutral, we quantify it with fuzzy values according to the adjectives' embodiment. We only give emotion values to the basic node adjectives in our net and the emotion values of deep node adjectives are obtained by emotion reference.

2 The Method

2.1 Our Viewpoint

Compared with the traditional view that concepts (nouns) are the basis of humans' cognition to the world, we hold that emotions are the core and motive power for human to know the new thing and whole world. Human beings contain these emotions in the descriptions (usually adjectives) of the things. Given a new thing, it is the descriptions which contain emotions that come up to humans' mind and then construct the concepts step by step. As to concepts, we view them as the attachments to adjectives (As shown in Figure 1).



Fig. 1. The central importance of emotions in humans' embodiment system and the relationship between emotions, adjectives and concepts(nouns)

2.2 Some Hypothesis

To construct such a knowledge net, we propose some hypothesis:

- 1) It is the adjectives that are the basis of humans' embodiment in knowing the world instead of concepts.
- 2) Humans' embodiment is from basic to esoteric. Originally, human beings know the world based on their embodied senses and then their imagination takes them to the deeper cognition.
- 3) Adjectives are the initial embodied descriptions to the new things and induce the appearance of concepts.
- 4) Emotions are contained in adjectives and they can be under the inference.

2.3 The Structure of the Net

According to the theory of embodiment, we consider that the five senses(i.e.tactile sensation, auditory sense, nose, sight, sense of taste) are the basis of embodiment. Thus, we take the adjectives of five senses as the nodes of our net. We employ the most direct sensory adjectives as the basic nodes of the net and expand them to the more esoteric ones as the second and third layers. This kind of expansion is based on the process that human beings knowing the new things from childhood to grow-up. We believe that the process is always deeper and deeper. Initially, when is a baby, a person cannot have any concept in his brain and given a "stone" as instance, the baby cannot know that it is called "stone".He touches it and feels *adamant* and *cold*, which are the first description of "stone".Then he constructs the concept of "stone" in his mind and then nouns work in the process of cognition.

As showed in Figure 2, we divide the net into three layers: the first layer contains humans' initial descriptions to the given things and they are the first embodied impressions in childhood; the second layer contains more abstract or deeper descriptions and infers that the development of humans' cognition contains some imagination ; the third layer contains the psychological descriptions based on humans' totally imagination and it is the final cognition to the world.

In the first layer, it contains humans' initial descriptions (e.g. Giving a stone to a baby, he must feel it "cold" and "hard" through his tactile sensation and constructs such impressions in his brain, and then he constructs the concepts of "stone", instead of knowing that "it is a stone" at the first time.). Thus, we consider that the adjectives come to one's mind firstly and then concepts,when given a new thing. The adjectives are the motive power for human beings to know the world.

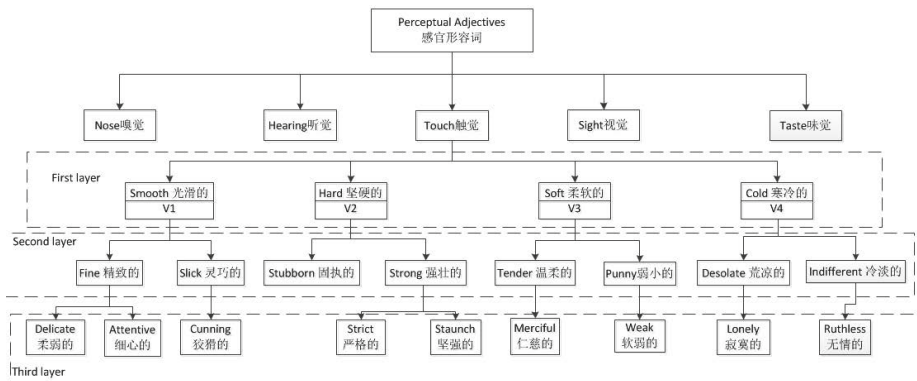


Fig. 2. Net of tactile sensation in five senses: the layer structure and parts of the adjective nodes. v_1-v_4 are the fuzzy emotion values given to the adjectives in first layer.

In the second layer, the adjectives are more abstract than the ones in the first layer and they are induced by the first layer with a little imagination in the process of humans’ cognition. As to the third layer, it is totally the production of humans’ imagination. Usually, the metaphors arise in these two layers. As showed, we give the fuzzy value of emotions to the adjectives in the first layer and the ones in the second and third can be induced by emotion inference.

Figure 2 is only a small subset of the total net, besides the sense of touch, the other four sense are also contained in our net.

2.4 Nouns as Attachments to the Net

Considering the effects of concepts in the process of humans’ cognition, even though we don’t take them as the key of this process, we consider that they can be attachments to the adjectives. Same as the emotions, we only give the concrete nouns and the abstract ones can be obtained by inference. As showed in Figure 3.

2.5 Applications and Expansibility

The first and major application of our knowledge net is metaphor comprehension. For instance, giving a metaphorical expression *Woman is water*. We can easily extract the properties of *woman* are ”tender , beautiful and so on” and the property of *water* are ”soft, cold, smooth and so on” . Apparently, in our net, *soft* is the father node of *tender*, which means the identification of this metaphor is that *Woman is tender*.

With this knowledge net, we realize the inference in emotion analysis of given text. Different from the existing approaches to divide emotions into six primitive types or simply positive, negative and neutral, we give them fuzzy values in the

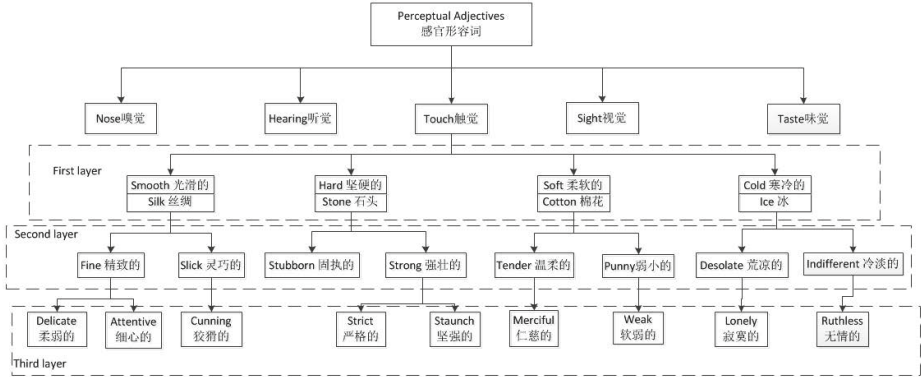


Fig. 3. Concepts as attachments to the adjective net

view of embodiment. The fuzzy values in this paper are not pure meaningless value, but contain semantics which can express the original cause and the development of emotions. Applying such fuzzy values makes it more explicit to emotion inference and emotion computing.

What is more, this knowledge net is not limited by language or culture. It can be expanded to different languages (e.g. Chinese, English), cultures and contexts.

3 Reliability Analysis

3.1 The Reliability of Layers Division

We using MRC Psycholinguistic Database Machine Usable Dictionary (Coltheart,1981) to test the consistency and reliability of our knowledge net. The MRC Psycholinguistic Database Machine Usable Dictionary (Coltheart,1981) includes 150837 words rated with the abstractness by human subjects in psycholinguistic experiments[10].The rating range from 158(highly abstract) to 670(highly concrete). We test the words in each layer of our net with the degree of MRC,and it indicated that the net agree with the human judgement of MRC. Figure 4 gives some examples,among with the numbers are the degree of each word in MRC.

3.2 The Reliability of Adjectives Expansion

We take a online visual thesaurus dictionary-*Thinkmap*[11] to test and verify the reliability of adjectives expansion in our knowledge net.*Thinkmap* is an interactive tool that allows users to discover the connections between words in a visually captivating display and works with more than 145,000 words and 115,000 meanings organized in an innovative and intuitive design. Figure 5 and Figure 6 show the expansion of "smooth" to "fine" and "slick" and then from "fine" and "slick" to "delicate" and "cunning"(which is marked by red dash).That is , the expansion of the adjectives in our net is rational and meaningful.

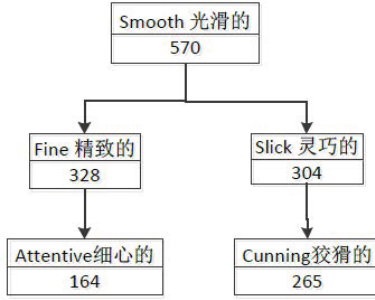
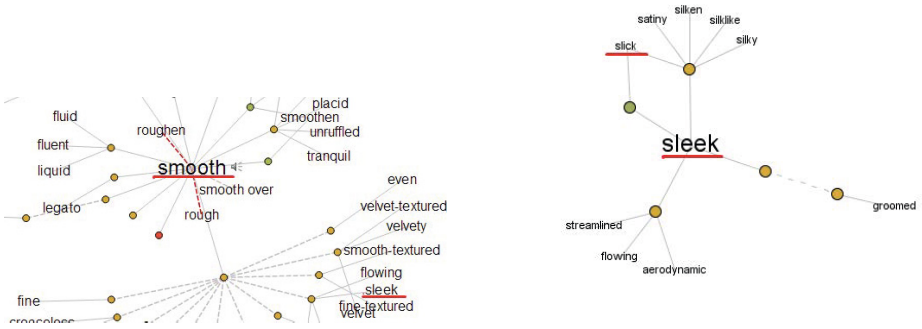
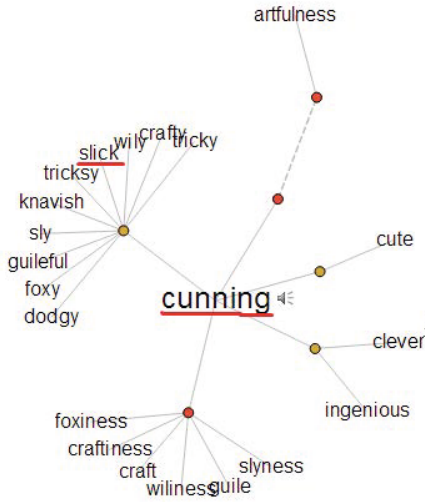


Fig. 4. Examples of the words in our net tested with the degrees in MRC



(a) The expansion from "smooth" to "slick"



(b) The expansion from "slick" to "cunning"

Fig. 5. An expansion from "smooth" to "slick" to "cunning" in the visual online thesaurus dictionary-Thinkmap

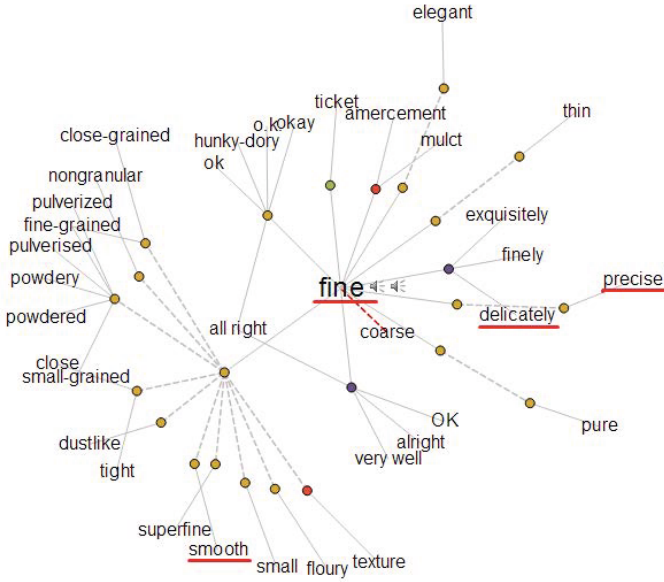


Fig. 6. An expansion from "smooth" to "fine" to "delicate" in the visual online thesaurus dictionary-*Thinkmap*

4 Conclusion and Future Work

We present an adjective-based knowledge net to face the metaphor comprehension, textual emotion analysis and emotion inference in this paper in a view of cognitive embodiment. Its application on metaphor comprehension does not employ any statistics method or hand-craft knowledge resource, but the embodied cognition to the new things of human beings. This kind of net is not a static knowledge net like WordNet or FrameNet, but a dynamic one which can be expanded with any different culture, emotion or any different context. In the part of emotion inference, we describe the emergent and development of emotions. Within inference, we realize the emotion analysis in text. What's more, we don't employ the existing method to divide the emotion into simply negative, positive or neutral, but qualify emotions with fuzzy values. Thus, emotions will be clearer and easier for us to analyze in the large text. In the future work, it is devoted to construct a metaphor evaluation system with this knowledge net and realize the emotion inference with deep excavation. Also, how to connect nouns with this net more closely is one of our future work.

Acknowledgments. Funding was provided by the National Natural Science Foundation of China under Grant(No. 61075058).

References

1. Lakoff, G., Johnson, M.: *Philosophy in the flesh: The embodied mind and its challenge to western thought*. Basic books (1996)
2. Langacker, R.W.: *Grammar and Conceptualization*. Mouton de Gruyter, Berlin (2000)
3. Lakoff, G.: *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. The University of Chicago Press, Chicago (1987)
4. Piero, S.: Book review of George Lakoff [DB/OL] (2004), <http://www.thymos.com/mind/Lakoff>
5. Turner, M.: *The Literary Mind*. OUP (1996)
6. Sriram, S., et al.: An enhanced approach for classifying emotions using customized decision tree algorithm. In: *2012 Proceedings of IEEE Southeastcon* (2012)
7. Seol, Y.-S., Kim, D.-J., Kim, H.-W.: Emotion Recognition from Text using Knowledge-based ANN. In: *The 23rd International Technical Conference on Circuit/Systems, Computers and Communications (ITS-CSCC-2008)* (2008)
8. Ma, C., Prendinger, H., Ishizuka, M.: *A Chat System Based on Emotion Estimation from Text and Embodied Conversational Messengers* Graduate School of Information Science and Technology, University of Tokyo and National Institute of Informatics, Japan
9. Holzman, L.E., Pottenger, W.M.: *Classification of Emotions in Internet Chat: An Application of Machine Learning Using Speech Phonemes*. Computer Science and Engineering, Lehigh University leh7, billp@lehigh.edu LU-CSE-03-002
10. <http://ota.oucs.ox.ac.uk/headers/1054.xml>
11. <http://www.visualthesaurus.com>

A Method of Density Analysis for Chinese Characters

Jingwei Qu, Xiaoqing Lu, Lu Liu, Zhi Tang, and Yongtao Wang

Institute of Computer Science & Technology of Peking University, Beijing, China
{qujingwei, lvxiaoqing, liulu.pku, tangzhi, wyt}@pku.edu.cn

Abstract. Density analysis plays an important role in font design and recognition. This paper presents a method of density analysis for Chinese characters. A number of density metrics are adopted to describe the density degree of a character from both local and global perspectives, including center-to-center distance of connected components, gap between connected components, ratio of perimeter and area, connected components area ratio, and area ratio of holes. The experiment results demonstrate that the proposed method is effective in measuring the density of Chinese characters.

Keywords: density analysis, shape analysis, porosity, compactness, connected components.

1 Introduction

Density is a significant factor in the design, recognition, and other applications of Chinese characters as fonts. Three scenarios illustrate how density metrics benefit the design and application of fonts. First, the evaluation of font beauty is related to density, as shown in Fig. 1. The density adjustment of a single character results in different visual effects.



Fig. 1. An example of different density of a same character in same style

Second, considering the aesthetic quality of a page, the density of different characters from the same font should not considerably vary, especially for Chinese characters whose stroke numbers differ significantly. In general, characters with many strokes tend to appear extremely dense on printed pages, whereas characters with relatively few strokes appear sparse. However, as Fig. 2 illustrates, some designs for a font without appropriate adjustments result in an unacceptable visual perceptions from the scale of a paragraph or a whole page.

Third, cross-cultural communication is becoming a staple to modern life at the present time, resulting in a large number of publications with mixed languages and characters. Density is an important factor for editors and designers to select fonts for different languages to achieve the consistent overall effect of layout.

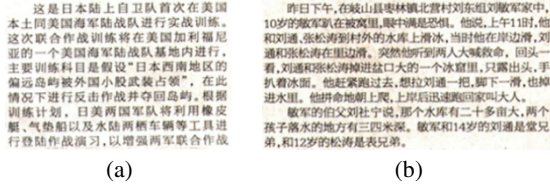


Fig. 2. Comparison between layout results of different designs for the same kind of fonts. (a) shows stronger consistency than (b) as considering more about density.

However, character density can only be estimated by human subjective judgments rather than by a quantification method in most font companies. At least three challenges exist. First, density degree is basically evaluated by human visual perception, which lacks reasonable visual models. Second, various factors influence shape density, thus requiring deep-seated shape analysis. Lastly, no evaluation method or common dataset is authorized to judge the validity of the density metric.

This paper describes a feasible method for calculating the density of Chinese characters. We regard Chinese characters as shapes with multiple connected components and analyze several factors influencing the density of the characters, such as center-to-center distance of connected components, gap between connected components, ratio of perimeter and area, scale of connected components, and scale of holes.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 introduces some basic concepts. Section 4 details the proposed method about the measurement of density. Section 5 shows the experiment results. Finally, Section 6 draws the conclusions of this study and recommends future work.

2 Related Work

Research on character density has so far been few. However, density analysis has been involved in some studies on pattern recognition and image retrieval, which use similar concepts such as porosity and compactness. Some of the related studies are briefly examined in this section.

Song et al. [2] explicitly defined porosity as the scale of the gap between connected components based on mathematical morphology [7]. A closing operation was adopted on the object shape using a circular element with a different radius, analyzing the scale of the gap with a multi-scale. Finally, porosity was described by the diameter of the minimum circle connecting all connected components.

Bribiesca [3] used compactness to separately analyze 2D and 3D shapes. An object in the 3D domain, for example, has compactness that is classically related to the enclosing surface area and volume and can be measured by the ratio $(\text{area}^3)/(\text{volume}^2)$.

Liu et al. [8] proposed a concept of foreground pixel density, which is simply defined as the ratio of foreground pixels in an object with respect to the foreground pixels in the entire image. Liu et al. [10] employed the density distribution feature, which is defined as a matrix whose components determine the relative density of the foreground pixel in each small region.

Liu et al. [11] presented point–line distance distribution (PLDD) to detect arbitrary triangles, regular polygons, and circles, based on the common geometric property that the in-center of the shape is equidistant to the tangential lines of the contour points. Unlike SC [12] and IDSC [13], PLDD directly presented image features that are very similar to density by distance distribution.

The above analysis shows that the density of shapes with multiple connected components has not been particularly and extensively studied as a separate issue. Studies are merely involved in some image research, and the methods have limitations. For example, porosity, defined as the gap between connected components, merely employs the minimum gap, which is a simple description of the density. Compactness only qualitatively represents dispersion of a shape without considering the quantification of distance between pixels. Thus the description of the density is not comprehensive.

3 The Basic Concepts

Before discussing the algorithm proposed in this study, some basic concepts about density are necessary to be introduced.

3.1 Convex Hull

The convex hull of a shape is defined as its initial envelope, as shown in Fig. 3.



Fig. 3. The effect of convex hull: (a) a Chinese character (b) corresponding convex hull

3.2 Hole

A hole of a connected component is a closed region whose gray level is obviously different from others. In a binary image, a hole is a black region in a white connected component, as shown in Fig. 4.



Fig. 4. The effect of filling holes: (a) a Chinese character (b) the effect after filling holes

3.3 Morphologic Closing Operation

A shape's morphologic closing operation corresponds to (1) the morphologic dilation and (2) the morphologic erosion with Eq. (1) – (6) illustrated in Gonzalez et al [26]:

$$A \cdot B = (A \oplus B) \ominus B \quad (1)$$

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (2)$$

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\} \quad (3)$$

$$A^c = \{w | w \notin A\} \quad (4)$$

$$\hat{B} = \{w | w = -b, b \in B\} \quad (5)$$

$$(B)_z = \{c | c = b + z, b \in B\} \quad (6)$$

4 Density Analysis

Density is one of the important properties for shape analysis. A character, in this paper, is regarded as a shape composed of multiple connected components. By analyzing the elements and structures in characters, many factors contributing to the density are found. We select five dominant metrics for a more in-depth analysis and effective description of density: (1) center-to-center distance of connected components, (2) gap between connected components, (3) ratio of perimeter and area (4) connected components area ratio, and (5) area ratio of holes. More details about them and their relationships are provided in following subsections.

4.1 Center-to-Center Distance of Connected Components (CCDCC)

To achieve an accurate description of density, the distribution of all connected components in a Chinese character is taken into account firstly. The distances between geometric centers of connected components can be used as an important feature. Hence, we propose the connected components center distance (CCDCC) to represent the layout information in a character. The CCDCC is defined as following:

$$d_{ij} = \sqrt{(\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2} \quad (7)$$

$$D = \{d_{12}, d_{13}, \dots, d_{ij}, \dots, d_{n(n-1)}\}, |D| = C(n, 2) \quad (8)$$

where (\bar{x}_i, \bar{y}_i) and (\bar{x}_j, \bar{y}_j) are the i^{th} and j^{th} connected component's center, respectively, $i = 1, 2, \dots, n, j = 1, 2, \dots, n, i \neq j$, D is the set of center distances, and n is the number of connected components of a Chinese character.

Referring to the perspective that Liu et al. [8] and Bai et al. [9] proposed graph structure for image matching, we represent each of the connected components with its centroid. Then, a Chinese character is converted to a graph, all connected components mapping to the graph nodes. CCDCC is based on distances between each pair of the nodes.

The CCDCC metric, obviously, is sensitive to the areas of connected components. When the area is small, it performs well in revealing the distance between two con-

nected components. On the contrary, if the regions of connected components enlarge in size, the distance between them is unsuitable to be described only with CCDCC. To eliminate the negative influence, we assign different weights to the CCDCCs, a minimal weight w_{min} given to the maximal center distance d_{max} , a maximal weight w_{max} given to the minimal center distance d_{min} , and an average weight w_{avg} given to the average center distance d_{avg} . The weighted CCDCC, WCCDCC, is obtained in Eq. (10),

$$d_{avg} = \frac{\sum d_{ij} - d_{max} - d_{min}}{|D| - 2} \quad (9)$$

$$WCCDCC = \frac{d_{max} * w_{min} + d_{min} * w_{max} + d_{avg} * w_{avg}}{L_{diagonal}} \quad (10)$$

where $\sum d_{ij}$ is the sum of all center distances and $L_{diagonal}$ is the diagonal length of bounding box of a character. Two examples are shown in Fig. 5.



Fig. 5. An example of WCCDCC (a) WCCDCC = 0.1348 (b) WCCDCC = 0.5142

4.2 Gap between Connected Components (GCC)

Another feature of component distribution is the gaps between connected components. We improve the porosity metric [2] based on gaps for more accurate density measurement at local level.

We adopt a closing operation to measure the gap between two connected components. The closing operation uses a predefined structure element (eg. circle) at different scales (radius) to analyze the size of the gap. A morphologic closing operation is adopted to maintain the original shape compared to the morphologic dilation, and it is useful for gap junction between connected components. The size of gap is obtained when the circle radius are large enough to merge the two connected components.

Therefore, the minimum diameter of the circle that leads to joining two connected components together reveals the size of gap objectively, and then, this diameter is defined as the size of gap (gap_{ij}) between two connected components. A set GAP_SET composed of all the gap_{ij} is obtained in Eq. (11) and Eq. (12) based on the operation of morphologic closing predefined in Section 3.3.

$$gap_{ij} = 2 * \min\{s | \text{Compn}(A_{ij} \cdot B_s) = 1\} \quad (11)$$

$$GAP_{SET} = \{gap_{12}, gap_{13}, \dots, gap_{ij}, \dots, gap_{n(n-1)}\}, |G| = C(n, 2) \quad (12)$$

where A_{ij} is a target shape composed of the i^{th} and j^{th} connected components, B_s is a predefined structure element (such as a circle) at scale s , $\text{Compn}(X)$ returns the number of connected components in the shape X , gap_{ij} is the gap between the i^{th} and j^{th}

connected component, $i = 1, 2, \dots, n, j = 1, 2, \dots, n, i \neq j$, and n is the number of connected components in a Chinese character. Finally, an average value of gaps is calculated with Eq. (13).

$$gap_{avg} = \frac{\sum gap_{ij} - gap_{most} * n_{most} - gap_{max} * n_{max} - gap_{min} * n_{min}}{|G| - n_{most} - n_{max} - n_{min}} \quad (13)$$

where gap_{most} , gap_{max} , gap_{min} and gap_{avg} respectively denotes the mode gap value, maximum value, minimum value and the average in the GAP_SET , n_{most} , n_{max} , n_{min} and n_{avg} denote the numbers of corresponding gaps individually.

4.3 Ratio of Perimeter and Area (RPA)

The appearance of the outline also affects character density. Bribiesca [3] proposed a simple and effective measurement for the compactness of 2D and 3D shapes. We employ compactness to depict the density of Chinese characters in reference to the equation in 2D domain. Compactness for a 2D shape associates the perimeter with the area of the shape and can be measured by the ratio (perimeter²) divided by area. The contact perimeter corresponds to the sum of the lengths of the segments shared by two adjacent pixels. The relation between the contact and the shape perimeters is represented by Eq. (14).

$$2P_c + P = 4LN \quad (14)$$

where P_c is the contact perimeter, P is the perimeter of the shape, L is the length of a side of the pixel, and N is the number of pixels. In Fig. 6, each square represents an image pixel, the solid lines denote the perimeter, and the dashed ones correspond to the contact perimeter, $N = 9, L = 1, P = 12, P_c = 12$.

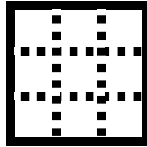


Fig. 6. A shape composed of 9 pixels

RPA of a Chinese character is calculated using Eq. (15).

$$C = \frac{n - P/4}{n - \sqrt{n}} \quad (15)$$

RPA can be applied not only to Chinese characters with multiple connected components, but also to those with single connected component. RPA reveals the distribution of all pixels of a character and partially describes its density. A higher pixel distribution results in a more concentrated image, as shown in Fig. 7.



Fig. 7. An example of the RPA (a) RPA = 0.9922 (b) RPA = 0.9514

4.4 Connected Components Area Ratio (CCAR)

The scale of connected components refers to the fullness by which the strokes of a Chinese character cover the entire region. A character is the foreground, and the convex hull is the background. The ratio of the foreground area, CCAR, is used for calculating the scale of the connected components. As shown in Eq. (16), CCAR is calculated as the ratio of the sum area of all connected components in a Chinese character with respect to the area of the region enclosed by the entire convex hull:

$$CCAR = \frac{s_1 + s_2 + \dots + s_i + \dots + s_n}{S_{con}} \quad (16)$$

where s_i and S_{con} denote the i^{th} area of the connected component and that of the convex hull in a Chinese character, respectively, $i = 1, 2, \dots, n$, and n is the number of connected components.

The higher the CCAR of a Chinese character, the denser is the image being perceived by human visual perception, as shown in Fig. 8.



Fig. 8. An example of CCAR: (a) CCAR = 0.2979 and (b) CCAR = 0.4122

CCAR can be applied to both characters with multiple connected components and characters with single connected components.

4.5 Area Ratio of Holes (ARH)

The structure of strokes in a Chinese character also affects the human vision about density. The hole, predefined in Section 3.2, is one of the most important and relatively dominant parts of a character. We consider the scale of the holes and define it as ARH based on the algorithm in Soffer et al. [4] with Eq. (17)

$$ARH = \frac{S_{hole}}{S_{com}} \quad (17)$$

where S_{hole} is the area of all holes in a Chinese character, and S_{com} is the area of the character whose holes have been filled.

Though CCAR can reveal density differences to a certain extent, it is not always the best factor. For example, the CCAR of Fig. 9(a) and (b) are nearly the same, but (b) looks denser than (a).



Fig. 9. An example of CCAR and ARH: (a) CCAR = 0.4171, ARH = 0.2195 and (b)CCAR = 0.4138, ARH = 0.0915

ARH reveals the relative size of the holes in a Chinese character. When ARH is high, the hole is large with respect to the character, and the character will seem loose and expanded to human vision. In the same example, Fig. 9 (a) has larger holes than (b), hence (b) appears denser.

4.6 Overall Density Metric

With the above extracted five density features, a density descriptor can be directly obtained by the five-dimension vector $V_{density}$ expressed as Eq. (18).

$$V_{density} = (WCCDCC, GAP, RPA, CCAR, ARH) \tag{18}$$

However, the five factors have varying importance. CCAR and RPA are only effective in local regions. They are used as weights to adjust other global factors. More specifically, to quantify the density of a character, we give different gaps obtain in section 4.2 with different weights. CCAR and RPA are adopted to calculate weights in Eq. (19) to enhance the gap value, because they have closer relationships with gaps,

$$GAP = gap_{most} * w_{most} + gap_{max} * w_{max} + gap_{min} * w_{min} + gap_{avg} * w_{avg} \tag{19}$$

where gap_{most} , gap_{max} , gap_{min} and gap_{avg} respectively denotes the mode gap value, maximum value, minimum value and the average; w_{most} , w_{max} , w_{min} and w_{avg} are their weights; GAP is the enhanced gap value, and an example is shown in Fig. 10.

We thus propose a combined form density metric. While GAP is calculated based on CCAR and RPA, the final density is the combination of the other three metrics, namely, ARH, WCCDCC, and GAP. We normalize GAP by $L_{diagonal}$ as Eq. (20) to derive GAP_{norm} . The overall density metric is defined in Eq. (21).

$$GAP_{norm} = \frac{GAP}{L_{diagonal}} \tag{20}$$

$$Density = (WCCDCC + GAP_{norm} + ARH) / 3 \tag{21}$$



Fig. 10. An example of GAP (a) GAP = 22 (b) GAP = 64

5 Experiments

5.1 Setup and Datasets

The experiment environment included an Intel Core i3 (3.30 GHz) processor with 4.00 G RAM and Windows 8 operating system, as well as Matlab 2013a.

We access four Chinese character databases, with each corresponding to a typeface, namely, Song, Fangsong, boldface, and regular script. Each database also contains 6,715 Chinese characters that are 128 x 128 in size.

5.2 Experiment Results

Comparison Results. We calculate the five features and the density of each Chinese character in the four databases and compare them. The samples of comparison results are shown in Table 1.

Table 1. The samples of comparison on the five features of four different typefaces













	Song	Fangsong	Boldface	Regular Script
(a)				
WCCDCC	0	0	0	0
GAP	2.5	2.5	2.5	2.5
RPA	0.8943	0.9306	0.9766	0.9619
CCAR	0.5399	0.9159	0.9690	0.8449
ARH	0	0	0	0
Density	0.0072	0.0079	0.0075	0.0073
(b)				
WCCDCC	0.3154	0.2938	0.3329	0.3150
GAP	8	8	8	12
RPA	0.9333	0.9111	0.9632	0.9353
CCAR	0.2874	0.2459	0.4413	0.3062
ARH	0.2595	0.2207	0.1381	0.1305
Density	0.2077	0.1883	0.1734	0.1734
(c)				
WCCDCC	0.3443	0.3017	0.3457	0.3295
GAP	57.7750	21.7273	59.1364	77.8000
RPA	0.9125	0.8940	0.9469	0.9238
CCAR	0.2845	0.2605	0.3913	0.3035
ARH	0	0	0	0
Density	0.2312	0.1456	0.2375	0.2751

Table 1. (continued)









(d)				
WCCDCC	0.2765	0.2792	0.2871	0.2895
GAP	23.7000	21.7000	26.4000	19.8000
RPA	0.9456	0.9352	0.9701	0.9564
CCAR	0.2888	0.2704	0.4118	0.3695
ARH	0.5340	0.5122	0.4197	0.4067
Density	0.3185	0.3113	0.2907	0.2764
(e)				
WCCDCC	0.3290	0.3129	0.3347	0.3096
GAP	49.0909	17.2364	55.2000	47
RPA	0.9211	0.8932	0.9491	0.9252
CCAR	0.3349	0.2620	0.4601	0.3263
ARH	0	0	0	0
Density	0.2715	0.1438	0.2346	0.143

Table 1 shows that a Chinese character has similar and different feature values relative to the typeface. For (a), when the number of the character strokes is small and the structure is simple, the density has no obvious differences between the four typical typefaces. However, from (b) to (e), as the number of strokes increases and the structure becomes more complex, the difference in the typefaces become more evident. Therefore, for most Chinese characters, density varies with different typefaces.

Clustering Results. We use the K-means algorithm to cluster Chinese characters with boldface based on the feature vectors comprising the five features, namely, WCCDCC, GAP_{norm} , RPA, CCAR and ARH. The experiment results demonstrate, that at $K = 3$, the differences between clusters are the most evident. Tables 2 and 3 summarize the experiment results and the corresponding character examples for all clusters.

Table 2. The clustering results based on the feature vectors

	WCCDCC	GAP_{norm}	RPA	CCAR	ARH
(a)	0.3383	0.4808	0.9519	0.4136	0.0732
(b)	0.2945	0.0725	0.9606	0.4276	0.1314
(c)	0.3161	0.2580	0.9543	0.4160	0.0776

Table 3. Some examples in each cluster

(a)		二		三		为		心	
		六		办		示		兰	
(b)		一		上		片		腱	
		髑		廛		幢		篝	
(c)		业		代		泯		粉	
		参		悴		臂		战	

As shown in Tables 2 and 3, the strokes of characters in cluster (a) are simple and are less compact compared with the other two clusters. Characters in cluster (b) are composed of two types: (1) single connected components characters and (2) characters with more complex strokes. Accordingly, the values of the features are either maximum or minimum among the three clusters and are most dense. The characters in cluster (c) do not have very evident differences. Some look dense, while others look less compact. The values of their features are the middle among the three clusters.

6 Conclusion and Future Work

This study focuses on the density of Chinese characters. We propose five metrics, namely, CCDCC, GAP, RPA, CCAR and ARH, to describe the density from the pixel, outline, and component levels. By combining the five metrics, both local and global information of the connected components are considered. The experiment results demonstrate that our method can not only discriminate density differences between different typefaces for the same Chinese character, but also depict density differences between characters with the same typeface. Future studies can explore more density factors to describe density and incorporate related knowledge such as psychology. In addition, we look forward to applying the density analysis to benefit more research fields.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant 61300061 and Beijing Natural Science Foundation under Grant 4132033.

References

1. Schomaker, L., de Leau, E., Vuurpijl, L.: Using Pen-Based Outlines for Object-Based Annotation and Image-Based Queries. In: Huijismans, D.P., Smeulders, A.W.M. (eds.) VISUAL 1999. LNCS, vol. 1614, pp. 585–592. Springer, Heidelberg (1999)
2. Song, J.-G., Lu, X.-Q., et al.: Envelope Extraction for Composite Shapes for Shape Retrieval. In: Proceedings of the International Conference on Pattern Recognition, Tsukuba Science City, Japan, pp. 1932–1935 (2012)
3. Bribiesca, E.: An Easy Measure of Compactness for 2D and 3D Shapes. *Pattern Recognition* 41, 543–554 (2008)
4. Soffer, A., Samet, H.: Negative Shape Features for Image Databases Consisting of Geographic Symbols. In: 3rd International Workshop on Visual Form Capri (1997)
5. Rosenfeld, A., Pfaltz, J.L.: Distance Functions on Digital Pictures. *Pattern Recognition* 1(1), 33–61 (1968)
6. Haralick, R.M.: A Measure for Circularity of Digital Figures. *IEEE Trans. Syst. Man Cybernet* 4(4), 394–396 (1997)
7. Hu, R.-X.: A Perceptually Motivated Morphological Strategy for Shape Retrieval. In: Huang, D.-S., Gan, Y., Gupta, P., Gromiha, M.M. (eds.) ICIC 2011. LNCS, vol. 6839, pp. 105–111. Springer, Heidelberg (2012)
8. Liu, L., Lu, Y., et al.: Near-duplicate Document Image Matching: a Graphical Perspective. *Pattern Recognition* (2013)
9. Bai, X., Yang, X.W., et al.: Learning Context-sensitive Shape Similarity by Graph Transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(5) (2010)
10. Liu, H., Feng, S.Q., Zha, H.B., et al.: Document Image Retrieval Based on Density Distribution Feature and Key Block Feature. In: Proceedings of International Conference on Document Analysis and Recognition, pp. 1040–1044 (2005)
11. Liu, H.M., Wang, Z.H.: PLDD: Point-lines Distance Distribution for Detection of Arbitrary Triangles, Regular Polygons and Circles. *J. Vis. Commun. Image R* (2013)
12. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Context. *IEEE Transactions Pattern Analysis and Machine Intelligence* 24(4), 509–522 (2002)
13. Ling, H.B., Jacobs, D.W.: Shape Classification Using the Inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(2), 286–299 (2007)
14. Guru, D.S., Nagendraswamy, H.S.: Symbolic Representation of Two-dimensional Shapes. *Pattern Recognition* 28, 144–155 (2007)
15. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock Graphs and Shape Matching. *International Journal of Computer Vision* 35(1), 13–32 (1999)
16. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of Shapes by Editing Shock Graphs. In: IEEE International Conference on Computer Vision, pp. 755–762 (2001)
17. Shi, C., Xiao, J., Jia, W., Xu, C.: Automatic generation of chinese character based on human vision and prior knowledge of calligraphy. In: Zhou, M., Zhou, G., Zhao, D., Liu, Q., Zou, L., et al. (eds.) NLPCC 2012. CCIS, vol. 333, pp. 23–33. Springer, Heidelberg (2012)
18. Marr, D., Vision, A.: A Computational Investigation into the Human Representation and Processing of Visual Information. Freeman and Company, WH San Francisco (1982)
19. Lai, P.K., Pong, M.C., Yeung, D.Y.: Chinese Glyph Generation Using Character Composition and Beauty Evaluation Metrics. In: International Conference on Computer Processing of Oriental Languages (ICCPOL), Honolulu, Hawaii, pp. 92–99 (1995)

20. Yu, K., Wu, J., Zhuang, Y.: Skeleton-based Recognition of Chinese Calligraphic Character Image. In: Huang, Y.-M.R., Xu, C., Cheng, K.-S., Yang, J.-F.K., Swamy, M.N.S., Li, S., Ding, J.-W. (eds.) PCM 2008. LNCS, vol. 5353, pp. 228–237. Springer, Heidelberg (2008)
21. Bribiesca, E.: Measuring 2-D Shape Compactness Using the Contact Perimeter. *Computers & Mathematics with Applications* 33(11), 1–9 (1997)
22. Bribiesca, E.: A Measure of Compactness for 3D Shapes. *Computers & Mathematics with Applications* 40(10), 1275–1284 (2000)
23. Bogaert, J., Rousseau, R., Van Hecke, P., et al.: Alternative Area-perimeter Ratios for Measurement of 2D Shape Compactness of Habitats. *Applied Mathematics and Computation* 111(1), 71–85 (2000)
24. Del Bimbo, A., Pala, P.: Image Indexing Using Shape-based Visual Features. In: Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, pp. 351–355. IEEE (1996)
25. Herrmann, P., Schlageter, G.: Retrieval of Document Images Using Layout Knowledge. In: Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 537–540. IEEE (1993)
26. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: *Digital Image Processing Using Matlab*. Princeton Hall Pearson Education Inc., New Jersey (2004)

Computing Semantic Relatedness Using a Word-Text Mutual Guidance Model

Bingquan Liu¹, Jian Feng¹, Ming Liu¹, Feng Liu¹,
Xiaolong Wang¹, and Peng Li²

¹ School of Computer Science and Technology, Harbin Institute of Technology,
Harbin, China, 150001

² School of Software, Harbin University of Science and Technology,
Harbin, China, 150080

Abstract. The computation of relatedness between two fragments of text or two words is a challenging task in many fields. In this study, we propose a novel method for measuring semantic relatedness between word units and between text units using an iterative process, which we refer to as the word-text mutual guidance (WTMG) method. WTMG combines the surface and contextual information when computing word or text relatedness. The iterative process can start in two different ways: calculating relatedness between texts using the initial relatedness of the words, or computing the relatedness between words using the initial relatedness of the texts. This method obtains the final relatedness result after the iterative process reaches convergence. We compared WTMG with previous relatedness computation methods, which showed that obvious improvements were obtained in terms of the correlation with human judgments.

Keywords: Semantic Relatedness, Mutual Guidance, Iterative Process, Initialization.

1 Introduction

The computation of semantic relatedness requires the estimation of the degree of association between two text fragments, which can be words, sentences, or documents (texts). For example, we may want to determine how two words are semantically related, such as *dog* and *cat*, or two pieces of text, such as *preparing a manuscript* and *writing an article*. Semantic relatedness measures have been applied in many natural language processing tasks such as information retrieval [4] and question answering [10].

Making judgments about the relatedness of different units is a common but complex task, which requires the surface meaning of the units and contextual knowledge about where they appear. Thus, we need to consider statistical information and the semantic information related to the words or texts. In this study, we introduce a new semantic relatedness computation model called the word-text mutual guidance model (WTMG). The mutual guidance between concept

A and concept B is defined as a process where A can be derived from B and B can be derived from A . In our model, we compute the relatedness among words using the text relatedness and the relatedness among texts can be calculated based on the word relatedness. We propose an iterative process that computes the relatedness among words and texts. Our model considers the semantic information obtained from a hierarchical lexical database such as WordNet and the statistical information contained in the corpus involved. The proposed method comprises two main steps. First, we establish the initial word relatedness or text relatedness and we construct a relatedness matrix. Second, the word relatedness and text relatedness are calculated iteratively.

The main contributions of this study are as follows. First, to exploit the information associated with words and texts, we propose the WTMG model to make full use of the internal relationships between words and texts. Second, we performed comparisons of many word and text semantic relatedness initialization methods. The remainder of this paper is organized as follows. We provide an overview of related work on word and text relatedness in Section 2. The WTMG model is described in detail in Section 3. Section 4 presents the experimental results and our conclusions are given in Section 5.

2 Related Work

Many methods have been proposed for semantic relatedness computation but they can generally be grouped into two categories: knowledge-based and corpus-based methods.

Knowledge-based methods, such as those of L&C [7], Wu&Palmer [13], Resnik [11], J&C [5], and Lin [8], employ information extracted from manually constructed lexical taxonomies, e.g., WordNet [1]. Previous studies have focused on developing appropriate measures while using WordNet as the primary knowledge source and they obtain relatively good results compared with corpus-based methods.

Corpus-based measures, such as LSA [6], ESA [2], SSA[3], employ probabilistic approaches to compute the semantic relatedness among words and texts. Most of these corpus-based measures map words or texts to the corresponding article in Wikipedia, which has emerged as a promising conceptual network for semantic relatedness computing in recent years.

However, these knowledge-based or corpus-based methods only consider semantic or statistical information, thus they ignore the fact that there must be relationships between a text and its component words. We propose the WTMG model to mine deeper relationship between words and texts. A similar approach was reported by [12], who aimed to calculate the short text similarity, but Wenyin's model has two drawbacks. First, it computes the initial word similarity by reconstructing WordNet, which is time consuming and unstable. Second, the model only selects the word relatedness to begin the iteration process and it ignores the initial text relatedness, which is also an important factor. Our model uses the most typical word relatedness computation method, which is available

directly with WordNet, to initialize the word relatedness matrix. Next, the initial text relatedness matrix can be calculated based on the initial word relatedness matrix, and the word and text relatedness are then calculated using an iterative algorithm.

3 WTMG Model

The proposed method comprises the following steps. Given a set of raw texts, the model first computes the initial relatedness between words using a knowledge-based method and the initial word relatedness matrix is then constructed. The text relatedness is computed based on the word relatedness matrix, and the word relatedness and text relatedness are then calculated iteratively until convergence. There is an alternative method for starting the iteration process, where we can calculate the initial text relatedness first and the word relatedness can then be calculated based on the initial text relatedness, but the two alternative methods both rely on an iterative process. Sections 3.1 and 3.2 introduce the typical initial word and text relatedness computation methods, respectively.

3.1 Word Relatedness Initialization

The measures used for computing the semantic relatedness belong to two categories.

Path-Based Measures. These measures compute the word relatedness as a function of the number of edges in the taxonomy along the path between two conceptual nodes c_1 and c_2 onto which the words w_1 and w_2 are mapped. The simplest path-based measure is the basic edge counting method, which defines the semantic distance as the number of nodes in the taxonomy along the shortest path between two concepts. The semantic relatedness is defined based on the semantic distance.

Information Content-Based Measures. [11] defined a criterion of similarity between two concepts as the extent to which they share common information. The information content is defined as $IC(c) = -\log P(c)$, where $P(c)$ is the probability that a randomly selected word in a corpus is an instance of concept c . Semantic relatedness between concepts are then calculated based on the information content.

In our study, we calculated the initial word relatedness with both path-based measures and information content-based measures to compare the performance of these approaches. Meanwhile, we need to ensure that the relatedness is computed between words with the same parts of speech. This is because most word-to-word knowledge-based measures cannot be applied across parts of speech, thus we added this restriction to all of the word-to-word relatedness measures.

3.2 Text Relatedness Initialization

An alternative method for initializing our WTMG model is calculating the text relatedness first. The typical approach finds the relatedness between two text

segments using the vector space model or latent semantic analysis. Although these methods are successful to some degree, these corpus-based relatedness methods cannot always identify the semantic relatedness of texts. For example, there is an obvious relatedness between the two text segments *I own a dog* and *I have an animal*, but most current text relatedness metrics fail with relatively short texts.

[9] proposed a method for measuring the semantic relatedness of texts by exploiting the information that can be extracted from the relatedness of their component words. The relatedness of two texts t_1 and t_2 is defined in Eq. (1).

$$sim(t_1, t_2) = \frac{1}{2} \left[\frac{\sum_{w \in \{t_1\}} (max(w, t_2) * idf(w))}{\sum_{w \in \{t_1\}} idf(w)} + \frac{\sum_{w \in \{t_2\}} (max(w, t_1) * idf(w))}{\sum_{w \in \{t_2\}} idf(w)} \right] \quad (1)$$

where $max(w, t)$ represents the maximum relatedness between w and component words of t . This method focuses on measuring the semantic relatedness of short texts by exploiting the deep relationships between words and texts, and combining the word relatedness to obtain the text relatedness, *COMB* in Table 2 shows the text relatedness results obtained using the word relatedness.

3.3 Iterative Procedure

We calculate the relatedness between words and texts in Sections 3.1 and 3.2, respectively, where both the word relatedness and text relatedness were calculated independently. In most cases, however, there are relationship among texts and words. Thus, if we want to compute the relatedness between *text1* and *text2*, the words in *text1* and *text2* can affect the relatedness between them, which is similar to Eq. (1). Normally, two texts may be similar if they share more co-occurring words. In addition, each word has synonyms, thus if two texts include synonymous information, they should be similar. Similarly, two words may share a common or similar concept if they co-occur in many texts or they appear in similar texts.

In general, the most straightforward method for calculating relatedness between two texts (e.g., t_p and t_q) is to use the text vector derived from the word-text matrix, which is defined by Eq. (2):

$$R(t_p, t_q) = sim(V(t_p), V(t_q)) \quad (2)$$

where $V(t_p)$ and $V(t_q)$ denote two text vectors formed by words.

Equation (2) is based on vector representation and many measurements are required to perform this calculation. Thus, we use *Cosine* as an example and Eq. (2) changes into Eq. (3):

$$R(t_p, t_q) = \sum_{k=1}^N (tf_{pk}) * (tf_{qk}) \quad (3)$$

where tf_{ij} is the frequency of word w_j in the i th text and N is the dimensionality of the feature space, or the total number of words that appear in the corpus.

In fact, texts are composed of words, thus if two texts share more topics with similar words, these two texts are relevant. This idea is useful when calculating the text relatedness based on the word relatedness. Thus, we expand the text relatedness to Eq. (4).

$$R(t_p, t_q) = \sum_{k=1}^N (tf'_{pk}) * (tf'_{qk}) \quad (4)$$

tf'_{pk} and tf'_{qk} can then be calculated using Eq.(5):

$$tf'_{pk} = \sum_{j=1}^N (tf_{pj}P_{jk}) \quad tf'_{qk} = \sum_{j=1}^N (tf_{qj}P_{jk}) \quad (5)$$

where P_{jk} indicates the normalized word relatedness, which can be calculated using Eq. (6).

$$P_{jk} = \frac{sim(w_j, w_k)}{\sqrt{\sum_{l=1}^N sim(w_j, w_l)^2}} \quad (6)$$

By incorporating guidance based on the word relatedness, Eq. (4) changes into Eq. (7).

$$R(t_p, t_q) = \sum_{k=1}^N \left[\left(\sum_{j=1}^N tf_{pj}P_{jk} \right) \left(\sum_{j=1}^N tf_{qj}P_{jk} \right) \right] \quad (7)$$

Similarly, the word relatedness can also be calculated by the vector represented by texts, which is defined in Eq. (8):

$$sim(w_p, w_q) = sim(V(w_p), V(w_q)) \quad (8)$$

where w_p and w_q denote two word vectors formed by texts.

We use *Cosine* as an example to compute the relatedness and Eq. (8) changes into Eq. (9):

$$sim(w_p, w_q) = \sum_{k=1}^M (tf_{kp}) * (tf_{kq}) \quad (9)$$

where M is the number of texts in the corpus. Two words may be more similar if they co-occur in many texts or they appear in similar texts. Based on this fact, Eq. (9) changes into Eq. (10):

$$sim(w_p, w_q) = \sum_{k=1}^M (tf'_{kp}) * (tf'_{kq}) \quad (10)$$

where tf'_{kp} and tf'_{kq} are defined by Eq. (11):

$$tf'_{kp} = \sum_{i=1}^M (tf_{ip}Q_{ik}) \quad tf'_{kq} = \sum_{i=1}^M (tf_{iq}Q_{ik}) \quad (11)$$

where Q_{ik} indicates the normalized text relatedness, which can be calculated using Eq. (12).

$$Q_{ik} = \frac{\text{sim}(t_i, t_k)}{\sqrt{\sum_{l=1}^M \text{sim}(t_i, t_l)^2}} \quad (12)$$

By incorporating guidance based on the text relatedness, the word relatedness can be computed using Eq. (13).

$$\text{sim}(w_p, w_q) = \sum_{k=1}^M \left[\left(\sum_{i=1}^M t f_{ip} Q_{ik} \right) \left(\sum_{i=1}^M t f_{iq} Q_{ik} \right) \right] \quad (13)$$

It is obvious that P_{jk} is derived from the relatedness between words and that Q_{ik} is derived from the relatedness between texts. We can see that the definitions of the relatedness between words and the relatedness between texts are cyclic. Thus, the relatedness between words and the relatedness between texts can be calculated using an iterative algorithm. Figure 1 shows the iterative process employed with WTMG. Two operations are repeated alternately, where one operation uses the word similarity to guide the text relatedness calculation and the other is the opposite. It is clear that the process can operate in two possible ways, which start from two initial points. The dotted line starts from $\text{Sim}^{(0)}(w_p, w_q)$ and the real line starts from $R^{(0)}(t_p, t_q)$. Thus, the process only requires the setting of one parameter: $\text{Sim}^{(0)}(w_p, w_q)$ or $R^{(0)}(t_p, t_q)$.

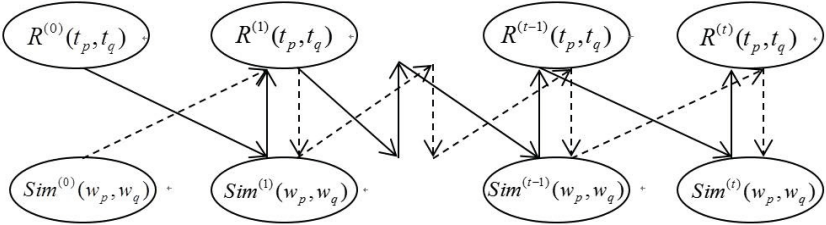


Fig. 1. Iterative Procedure of WTMG

The iterative process can be defined using Eqs. (14) and (15).

$$\text{sim}^{(t+1)}(w_p, w_q) = \sum_{k=1}^M \left[\left(\sum_{i=1}^M t f_{ip} Q_{ik}^{(t)} \right) \left(\sum_{i=1}^M t f_{iq} Q_{ik}^{(t)} \right) \right] \quad (14)$$

$$R^{(t+1)}(t_p, t_q) = \sum_{k=1}^N \left[\left(\sum_{j=1}^N t f_{pj} P_{jk}^{(t+1)} \right) \left(\sum_{j=1}^N t f_{qj} P_{jk}^{(t+1)} \right) \right] \quad (15)$$

Equations (14) and (15) show that the process begins with $R^{(0)}(t_p, t_q)$, and the process can also start in another way.

To verify the effectiveness of WTMG, we selected five similar text pairs (designated as S1 to S5) and five dissimilar text pairs (designated as U1 to U5). We applied our WTMG model to this small corpus example and the results are shown in Figures 1 and 2. Figure 1 shows the text relatedness results obtained after starting from $R^{(0)}(t_p, t_q)$ with WTMG, whereas Figure 2 shows the text relatedness results obtained after starting from $Sim^{(0)}(w_p, w_q)$ with WTMG.

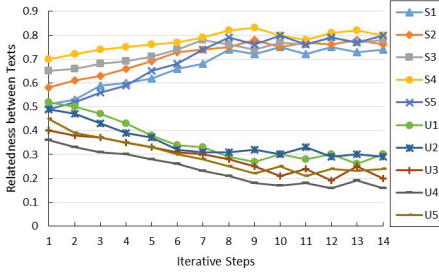


Fig. 2. Initialize Text Relatedness

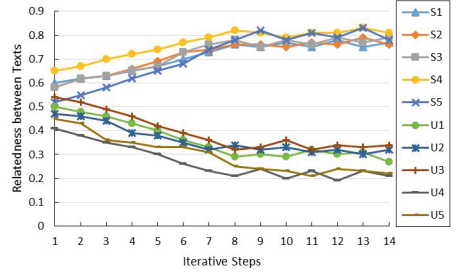


Fig. 3. Initialize Word Relatedness

Figures 2 and 3 demonstrate that our model obtained reasonable results, where it allowed the relatedness between similar texts to be greater, and the relatedness between dissimilar texts to be smaller. The results calculated by starting from the initial word relatedness or initial text relatedness were also similar. However, this "convergence" does not mean that the process converge to specific values and it simply refers to a balanced condition, where the values fluctuated in a small range repeatedly. To make the iterative process converge, we add a damping factor λ and λ is set by a kernel function that regresses as time passes. By minimizing λ , Eqs. (14) and (15) can finally converge. After adding the damping factor λ , the iterative equations change into Eqs. (17) and (16).

$$sim^{(t+1)}(w_p, w_q) = (1 - \lambda)sim^{(t)}(w_p, w_q) + \lambda \sum_{k=1}^M \left[\left(\sum_{i=1}^M t f_{ip} Q_{ik}^{(t)} \right) \left(\sum_{j=1}^M t f_{jq} Q_{jk}^{(t)} \right) \right] \quad (16)$$

$$R^{(t+1)}(t_p, t_q) = (1 - \lambda)R^{(t)}(t_p, t_q) + \lambda \sum_{k=1}^N \left[\left(\sum_{j=1}^N t f_{pj} P_{jk}^{(t+1)} \right) \left(\sum_{j=1}^N t f_{qj} P_{jk}^{(t+1)} \right) \right] \quad (17)$$

λ can be set between 0 and 1, and t represents the t th iteration. Theoretically, λ can be different in Eqs. (16) and (17), but we used the same value of λ in our experiments for simplicity. Theoretically, the convergence of relatedness cannot be guaranteed, thus we decrease λ by 20% during each iteration in practice.

3.4 Time Complexity Analysis

After matrix Q and P have been calculated, we can calculate the relatedness. the process that Eq. (7). and Eq. (13) show can be written as:

$$R_t = TQQ'T' \quad (18)$$

$$S_w = T'PP'T \quad (19)$$

where T is the matrix of terms in text, Q and P are regularizations of S_w and R_t . T multiply Q whose time complexity is $O(kn^2)$, so the time complexity of Eq.(18) is $O(kn^2 + k^2n)$, similarly Eq.(19) is $O(k^2n + kn^2)$. Suppose after t step, the model has converged, so the time complexity of calculating P is $O(tkn^2 + tk^2n)$, similarly Q is $O(tk^2n + tkn^2)$. Matrix T , Q , P are sparse matrix, and the size of nonzero number were written as L_t , L_q , L_p . As the complexity of multiplying between matrix and vector is linear, the complexity of TQ can be declined as $O(kL_q)$ and the complexity of Eq.(18) can be written as $O(kL_q + kL_t)$, similarly Eq.(19) as $O(nL_p + nL_t)$. So the time complexity for calculating matrix P and Q are $O(tkL_q + tkL_t)$ and $O(tnL_p + tnL_t)$. Thus the time complexity of the algorithm is $O(tkL_q + tkL_t + tnL_p + tnL_t)$

4 Experiments

In our experiments, parameter λ was set to 0.5 and our model was applied using two alternative methods. First, we started our model based on the initial word relatedness, and the text relatedness and word relatedness were calculated iteratively until convergence, which we denote by *WTMGW*. Second, we initialized the text relatedness first, and the word relatedness and text relatedness were computed iteratively until convergence, which we denote by *WTMGT*.

The word relatedness measures use WordNet as a resource and the results shown in Table 1 indicate that the *Lin* measure performed the best among all the path-based and information content-based measures. Thus, we used the *Lin* method to calculate the word-to-word relatedness in *COMB*. In our *WTMG* model, we used the *Lin* measure to initialize the word relatedness and the text relatedness was initialized with the *COMB* measure, i.e., *WTMGW* and *WTMGT* respectively.

We used several standard word-to-word and text-to-text datasets to evaluate the representation strength of our mutual guidance semantic relatedness model. Correct correlations are typically used to evaluate the semantic relatedness, thus we used *Pearson's correlation coefficient* γ and *Spearman's rank correlation coefficient* ρ in our study, both of which are important for semantic relatedness evaluations.

4.1 Word Relatedness

To evaluate the effectiveness of the *WTMG* model in determining the word-to-word relatedness, we employed three standard datasets that have been used widely in previous studies.

Rubenstein and Goodenough(RG65) comprises 65 word pairs that range from synonymy pairs (e.g., *car-automobile*) to completely unrelated terms (e.g., *noon-string*). The 65 noun pairs were annotated by 51 human subjects. All of the noun pairs are non-technical words and they are scored using a scale from 0 (not related) to 4 (perfect synonymy).

Miller-Charles(MC30) is a subset of the Rubenstein and Goodenough dataset that comprises 30 word pairs. The relatedness of each word pair was rated by 38 human subjects using a scale from 0 to 4.

WordSimilarity-353(WS353) is known as Finkelstein-353 and it comprises 353 word pairs annotated by 13 human experts using a scale from 0 (unrelated) to 10 (very closely related or identical). The Miller-Charles set is a subset of the WordSimilarity-353 dataset. Unlike the Miller-Charles dataset, which only contains single generic words, the WordSimilarity-353 set also includes phrases (e.g., "Wednesday news"), proper names, and technical terms, thus it presents an additional degree of difficulty for any relatedness metric.

Mturk-771(MT771) comprises 771 English word pairs and with their mean relatedness scores. The scores were collected using the Amazon Mechanical Turk and at least 20 ratings were collected for each word pair, where each judgment task comprised a batch of 50 word pairs. The ratings were collected on a scale of 1 to 5, where 5 denotes highly related and 1 denotes not related. The relatedness value of each word pair was the mean score given by the users.

We used the *Reuters News*¹ dataset when calculating the word relatedness, which is available on the Web. It contains 10,788 documents and approximately 130 million words. We used this large dataset to calculate the word relatedness and text relatedness iteratively because we needed a corpus that would include all the word pairs found in the standard datasets described above.

Table 1. Pearson and Spearman results for the word relatedness datasets

Method	Pearson(γ)				Spearman(ρ)				
	MC30	RG65	WS353	MT771	MC30	RG65	WS353	MT771	
Knowledge-based	Wup	0.778	0.784	0.282	0.477	0.750	0.755	0.339	0.398
	J&C	0.695	0.731	0.354	0.498	0.820	0.804	0.318	0.402
	L&C	0.779	0.839	0.313	0.503	0.768	0.797	0.302	0.410
	Lin	0.835	<u>0.858</u>	0.329	0.513	0.750	0.788	0.348	0.424
	Resnik	0.813	0.836	0.362	0.431	0.693	0.731	0.353	0.404
Corpus-based	LSA	0.725	0.644	0.563	–	0.662	0.609	0.581	–
	ESA	0.588	–	0.503	–	0.727	–	0.629	–
	SSA	0.778	0.850	0.590	–	<u>0.843</u>	0.800	0.537	–
Ours	WTMGW	0.879	0.861	0.622	0.572	0.846	0.826	0.750	0.480
	WTMGT	<u>0.871</u>	0.847	<u>0.602</u>	<u>0.539</u>	0.820	0.830	<u>0.748</u>	<u>0.477</u>

Table 1 shows the results obtained using our mutual guidance model compared with the state-of-the-art methods (knowledge-based and corpus-based). The left

¹ <http://about.reuters.com/researchandstandards/corpus/>

column in Table 1 shows the measurement methods. The results in bold indicate the best results for a dataset and underlining denotes the second best results. Table 1 shows that the knowledge-based methods obtained very good results with the *MC30* and *RG65* datasets, which can be explained by the deliberate inclusion of familiar and frequently used dictionary words in these sets. As expected, our model performed better with large datasets such as *WS353* (γ from 0.282 to 0.622 and ρ from 0.302 to 0.750) and *MT771* (γ from 0.431 to 0.572 and ρ from 0.398 to 0.480), probably because the large datasets contained more technical and culturally biased terms, which cannot be covered by knowledge-based measures.

We can also conclude from Table 1 that our proposed model *WTMGW* performed best with most of the datasets, followed by *WTMGT*. Clear, the results obtained with *WTMGW* and *WTMGT* are similar because they only differed in terms of their beginning points. They shared the same iterative process, thus the results were similar after several iterations. It is also interesting to note that the performance of *WTMGW* was superior to that of the *SSA* method, although *SSA* uses Wikipedia as its knowledge resource. This may be because Wikipedia is very complicated and it contains a high level of noisy information, whereas our model only utilizes the context information, which is reliable and the semantics are abundant.

4.2 Text Relatedness

To evaluate the effectiveness of the *WTMG* model in determining the text-to-text relatedness, we used two datasets that have been employed in previous studies.

Lee50 comprises 50 documents collected from the Australian Broadcasting Corporation’s news mail service. Each document was scored by ten annotators based on their semantic relatedness to all the other documents. The user annotations were then averaged per document pair, thereby yielding 2,500 document pairs and their similarity score annotations. We found that there were no significant differences between the annotations when order of the documents in a pair differed, thus the evaluations used only 1225 document pairs after ignoring duplicates.

Li30 is a sentence pair similarity dataset, which was obtained by replacing each of the Rubenstein and Goodenough word-pairs with their respective definitions in the Collins Cobuild dictionary. Each sentence pair was scored by 32 native English speakers and the scores were averaged to generate a single relatedness score per sentence pair. The scores were skewed toward low similarity sentence-pairs, so a subset of 30 sentences was selected manually from the 65 sentence pairs to maintain an even distribution across the similarity range.

AG400 is a domain-specific dataset related to computer science, which is used to evaluate the semantic relatedness of real-world applications such as short answer grading. The original dataset comprises 630 student answers and their corresponding questions. Each answer was graded by two judges on a scale from 0 to 5 and the Pearson’s correlation coefficient between human judges was found to be 0.64. We noted a large skew in the grade distribution toward the high

Table 2. Pearson and Spearman results for the text relatedness datasets

Method	Pearson(γ)			Spearman(ρ)			
	Li30	Lee50	AG400	Li30	Lee50	AG400	
Knowledge-based	COMB	0.810	<u>0.702</u>	0.480	0.832	0.356	0.365
	VSM	0.759	0.639	0.386	0.773	0.289	0.304
Corpus-based	LSA	0.810	0.635	0.425	0.812	0.437	0.389
	ESA	0.838	0.696	0.365	0.863	0.463	0.318
	SSA	0.848	0.684	0.567	0.832	<u>0.480</u>	0.495
Ours	WTMGW	0.886	0.724	0.602	0.878	0.488	<u>0.486</u>
	WTMGT	<u>0.872</u>	0.673	<u>0.584</u>	<u>0.870</u>	0.452	0.512

end of the grading scale, thus we randomly eliminated 230 of the highest grade answers to obtain more normally distributed scores.

Table 2 shows the semantic relatedness results obtained with the text datasets, where WTMG was compared with the knowledge-based and corpus-based methods. The results show that *WTMGW* obtained very good results with *Li30* ($\gamma=0.886$ and $\rho=0.878$) and *Lee50* ($\gamma=0.724$ and $\rho=0.488$). It is also interesting to note that large improvements were obtained using *WTMGW* ($\gamma=0.602$, $\rho=0.486$) and *WTMGT* ($\gamma=0.584$, $\rho=0.512$) compared with *LSA*, *ESA*, and *SSA* based on evaluations with the AG400 dataset.

As shown in Table 2, *WTMGW* and *WTMGT* clearly delivered the best performance, and they provided great improvements with the *AG400* dataset, but they were only slightly better than other methods with relatively small datasets.

5 Conclusions

The existing methods used to measure semantic relatedness consider the knowledge and the corpus independently, and they ignore the internal relationships between words and texts. In this study, we developed a word-text mutual guidance model to mine the deep relationships between words and texts, which combines semantic and statistical information using an iterative process. The initial word relatedness is calculated based on WordNet, which is semantically richer than corpus-based approaches. The text relatedness is then calculated based on the word relatedness, where this process utilizes the relationship between a text and its component words in an effective manner. The experimental results demonstrated that our proposed model is more effective than the state-of-the-art methods for semantic computing. The evaluations using standard word-to-word and text-to-text relatedness benchmarks confirmed the superiority and consistency of our model.

However, the model remains time consuming and there is still room for improvement, e.g., it may be possible to optimize the algorithm using dimensionality reduction. In future work, we will apply this semantic relatedness model to other NLP tasks such as text clustering or relationship classification.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No.61100094, 61272383, 61300114 and 61103149). We thank the anonymous reviewers for their insightful comments.

References

1. Fellbaum, C.: WordNet. Wiley Online Library (1999)
2. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. *IJCAI* 7, 1606–1611 (2007)
3. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: *AAAI* (2011)
4. Jain, V., Singh, M.: Ontology based information retrieval in semantic web: A survey. *International Journal of Information Technology & Computer Science* 5(10) (2013)
5. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint [cmp-lg/9709008](https://arxiv.org/abs/cmp-lg/9709008) (1997)
6. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. *Discourse Processes* 25(2-3), 259–284 (1998)
7. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. *WordNet: An Electronic Lexical Database* 49(2), 265–283 (1998)
8. Lin, D.: An information-theoretic definition of similarity. In: *ICML*, vol. 98, pp. 296–304 (1998)
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: *AAAI*, vol. 6, pp. 775–780 (2006)
10. Moreda, P., Llorens, H., Saquete, E., Palomar, M.: Combining semantic information in question answering systems. *Information Processing & Management* 47(6), 870–885 (2011)
11. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint [cmp-lg/9511007](https://arxiv.org/abs/cmp-lg/9511007) (1995)
12. Wenyin, L., Quan, X., Feng, M., Qiu, B.: A short text modeling method combining semantic and statistical information. *Information Sciences* 180(20), 4031–4041 (2010)
13. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pp. 133–138. Association for Computational Linguistics (1994)

Short Text Feature Enrichment Using Link Analysis on Topic-Keyword Graph

Peng Wang, Heng Zhang, Bo Xu, Chenglin Liu, and Hongwei Hao

Institute of Automation of the Chinese Academy of Sciences
95 Zhongguancun East Rd.
Beijing, 100190 P.R. China

{peng.wang, heng.zhang, boxu, chenglin.liu, hongwei.hao}@ia.ac.cn

Abstract. In this paper, we propose a novel feature enrichment method for short text classification based on the link analysis on topic-keyword graph. After topic modeling, we re-rank the keywords distribution extracted by biterm topic model (BTM) to make the topics more salient. Then a topic-keyword graph is constructed and link analysis is conducted. For complement, the K-L divergence is integrated with the structural similarity to discover the most related keywords. At last, the short text is expanded by appending these related keywords for classification. Experimental results on two open datasets validate the effectiveness of the proposed method.

Keywords: topic Model, Short Text, Feature Enrichment, Link Analysis, SimRank, K-L divergence.

1 Introduction

With the advent of the era of big data, mass of short texts have been generated on the web and mobile applications, including search snippets, micro-blog, products review, and short messages. The classification of these short texts plays an important role in understanding user intent, question answering and intelligent information retrieval [1, 2]. Currently, how to acquire the effective representation of short text and enhance the categorization performance have been an active research issue and have drawn much attentions [3, 4].

Short text bring about data sparsity and ambiguity problems because they cannot provide enough word co-occurrence or contextual information [3]. Thus, the general text mining methods based on bag of words cannot apply directly to short texts because they ignore the semantic relations between words [2]. Moreover, some essentially related short texts may have very little overlapping keywords, which seriously affects the similarity measurement and the categorization performance [5].

To solve these problems, some methods have been proposed to expand the representation of short text using latent semantics or related words. The expansion information is derived from the training datasets internally [6], or from external corpus such as Wikipedia [3]. Zhang et al. [5] proposed a graph-based text similarity measurement and exploited background knowledge from Wikipedia to find semantic

affinity between documents. Phan et al. [4] presented a general framework to expand the short and sparse text by appending topic names discovered using latent Dirichlet allocation (LDA) [7].

With Search Engine, Sun A. [1] proposed a simple method for short text categorization by selecting the most representative words as query to search a few of labeled samples, and the majority vote of the search results is the predictable category. Sahami and Heilman [8] enrich the text representation by web search results using the short text segment as a query.

In this paper, we propose a method using link analysis on topic-keyword graph for enriching the feature representation of short text to overcome the sparsity and semantic sensitive problems. First, we apply the biterm topic model (BTM) [9] to extract topics. Then, we re-rank the keywords distribution under each topic according to an improved TFIDF-like score [10]. Finally, a topic-keyword graph is constructed to prepare for link analysis.

On the topic-keyword graph, we use the link analysis algorithm—SimRank [11,12] to compute the structural similarity between every two nodes. Further, the K-L divergence of topics distribution is integrated with the structural similarity to discern the most similar keywords. The short text is expanded by appending these similar keywords for classification. Our method can avoid noise (class irrelevant) words and extract salient keywords by synthesizing the semantic knowledge and link structure information. Experiments are conducted on search snippets and 20Newgroups to validate the effectiveness of the method proposed.

The rest of this paper is organized as follows. Section 2 briefly reviews the relevant works. Section 3 introduces our method of short text representation enrichment. Section 4 presents the experimental results on two open datasets. Finally, concluding remarks are offered in Section 5.

2 Related Work

Short text classification is a challenge due to its noise words, lacking of sufficient contextual information and the semantic sensitive problem [3]. Thus, traditional statistical methods usually fail to achieve satisfactory classification performance [2].

In recent years, some research focuses on how to utilize large-scale external data to explore semantic information, and enrich the original text to help text mining [13]. Gabrilovich and Markovitch [14] proposed a method to improve text classification performance by enriching document representation with Wikipedia concepts. Zhang et al. [5] presented a graph-based text similarity measurement using SimRank, which is the most related work to our study. Based on the background knowledge from Wikipedia, they build a document-concept bipartite graph to compute the affinity of different documents and then perform text classification.

SimRank [11] is applicable to any domain with node-to-node relationships for measuring the similarity between nodes. The motivation of SimRank is that two nodes are similar if they are related by similar nodes. Due to the latent semantics and link structure information embedded in the graph representation, the SimRank algorithm can be used as a similarity measurement on the semantic graph.

Based on consistent Wikipedia corpus, Phan et al. proposed a method to discover hidden topics using LDA and expand the original text [4]. Zhu et al. [15] proposed to use multi-original external corpus to model topics for better categorization performance. This method can draw more broad and accurate topics compared to that based on one external corpus. Chen et al. [3] pointed out that leveraging topics at multiple granularity can model short texts more precisely.

In order to overcome the insufficiency of LDA in modeling short text in terms of the document-level word co-occurrence, Yan et al. [9] presented a new variant of topic model—bi-term topic model (BTM), for extracting topics using bi-terms existing in the whole training dataset instead of document-level, which can well alleviate the problem of sparsity.

Unlike the probabilistic formulation topic model [7, 16] with the constraints that the admixture proportions of topics and likelihood function should be normalized, Zhu and Xing [17] presented a non-probabilistic one named sparse topical coding (STC), which can control the sparsity of inferred result directly. STC model achieved the state-of-the-art classification accuracy on 20Newgroups dataset.

3 Short Text Feature Enrichment Method

Our method is aimed to identify the topical-indicative words automatically from training dataset, and expand the short text for classification. Based on BTM and SimRank, a unified feature enrichment method named SRBTM is proposed.

3.1 Overview

After topic modeling, the keywords under each topic are used to produce a thesaurus by re-ranking algorithm. Then, we construct a topic-keyword bipartite graph and compute the similarity between keywords using link analysis combined with K-L divergence. Finally, these most related keywords are selected to expand the original short text. The detailed framework is shown in Figure 1.

According to the thesauri produced in re-rank stage, we search seed words appeared in the short text, and all keywords mentioned in the thesauri are used as candidate words. Then, we present a novel similarity measurement in Equation (1) to compute the affinity between seed words and candidate words. For each seed word, we select top- v candidate words with maximum $CScore$ to enrich the short text.

$$CScore(sw_i, cw_j) = \frac{SR(sw_i, cw_j)}{KL(sw_i, cw_j)}, \quad (1)$$

$SR(sw_i, cw_j)$ is the SimRank score between seed word i and candidate word j , and $KL(sw_i, cw_j)$ is the K-L divergence.

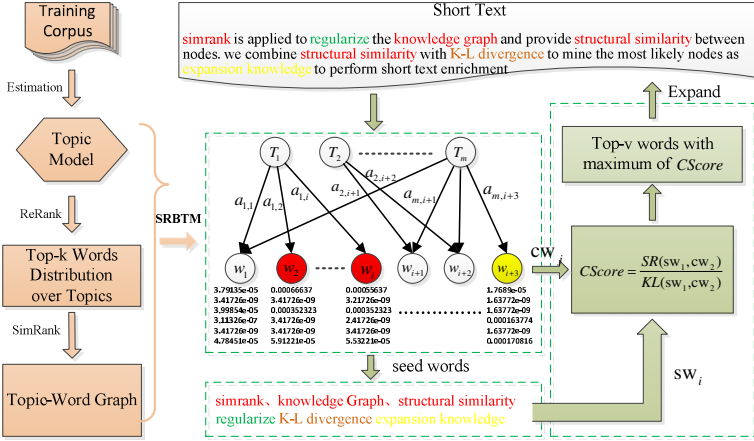


Fig. 1. The framework based on topic-keyword graph and link analysis

The merits of our unified framework are: (1) adding the most related topic keywords to short text can resolve the sparsity problem, and we can alleviate the synonymy and polysemy problems meanwhile; (2) the added keywords are new features for original short text which can be leveraged for categorization. Thus, topic related texts which do not share common seed words literally may be enriched with the same keywords. Then, the computation of similarity on these texts can be improved and classification performance will be enhanced; (3) BTM model short text at corpus-level, which reflects the global information. However, the keywords are added according to the content of each short text, which exploit the local information. So we synthesize the global information and local information to guarantee the effective enrichment.

3.2 Biterm Topic Model

With respect to topic extraction, documents are modeled as mixtures of latent topics and each topic can be further represented by a set of keywords. To overcome the data sparsity problem, Yan et al. proposed the biterm topic model (BTM) [9] especially for short text, which directly model the word co-occurrences in the whole corpus to make full use of the global information. The parameters and variables used in BTM are listed in Table 1.

Briefly, BTM as one of the improved variants of LDA, can effectively alleviate the data sparsity in modeling short text. Moreover, as demonstrated in Table 2 that the re-ranked keywords from each topic are high related, BTM make the representation of short text more topic-focused.

Table 1. Variables in BTM

Para.	Details
M	number of bi-terms
α, β	Para. for Dirichlet
$\bar{\theta}$	topic distribution
z	index of a topic
$\bar{\varphi}_{iz}$	i th word distribution
V	vocabulary size
K	the number of topics
Φ	a $K \times V$ matrix
BT	corpus with M biterns

Table 2. Most likely words of some topics

Topic0:	music band rock album song songs released
Topic1:	species food animals animal plants humans
Topic2:	energy mass field quantum particles force
Topic3:	india indian hindu pakistan sanskrit century
Topic4:	blood body brain heart cells muscle syndrome
Topic5:	water carbon oil chemical gas process oxygen
Topic6:	government party president constitution
Topic7:	power energy solar electric electrical
Topic8:	system data code software computer
Topic9:	horse opponent horses body hand match
Topic10:	south africa united country islands world

3.3 Re-ranking Method

The keywords extracted by BTM are denoted as $\Phi = \{\bar{\varphi}_z\}$. $\bar{\varphi}_z = [\varphi_{z,1}, \varphi_{z,1}, \dots, \varphi_{z,V}]$ is the word distribution vector with the length of V under topic z . Song et al. [10] presented a keyword re-ranking algorithm for LDA-based topic modeling. Word distribution $\Phi = \{\bar{\varphi}_z\}$ is applied to compute a TFIDF-like score for each keyword, and the original order of keywords is re-ranked.

To better identify salient information, we improve the re-ranking method in [10] for ranking BTM-derived keywords to refine the topic definitions. Different from the method in [10], we propose to use $\exp(\varphi_{z,i})$ as frequency-like term to compute the saliency score in (2),

$$SAS = \frac{e^{\varphi_{z,i}}}{\sum_{m=1}^M e^{\varphi_{z,i}}}, \quad (2)$$

where $\varphi_{z,i}$ is the probability distribution of the i th word under topic k . Then, the re-ranked results demonstrate in Table 2. Each line is a thesaurus corresponding to a specific topic which forms a clique in Figure 2, which show that our proposed method can make each topic more salient.

However, Figure 2(b) shows that there are still some keywords keep relevant to more than one topic. For short text expansion, these keywords may be noisy rather than be useful in providing discriminative information.

3.4 Topic-Keyword Graph Construction

After re-ranking the keywords for each topic, we construct a topic-keyword semantic graph as shown in Figure 2 and Figure 3. The semantic graph is built using topics and keywords as nodes, which are derived from BTM. Specifically, the hub node (or parent node) of each clique refers to a topic name. The top-k keywords in the thesaurus under a specific topic are selected as leaf nodes to form the clique. Then, all the

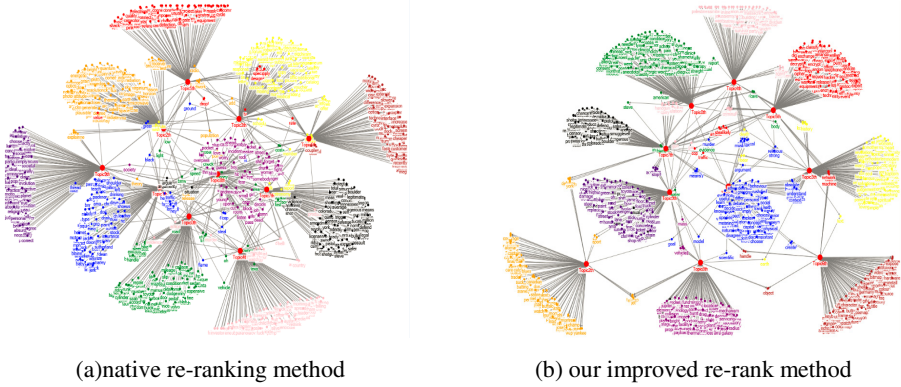


Fig. 2. Topic-Keyword Graph

cliques constitute the topic-keyword graph, which synthesize the semantic knowledge and link structure information.

The semantic graph is weighted by the score computed previously for keywords re-ranking using Equation (2). The weights are put on the edge from the hub nodes to leaf nodes as shown in Figure 3, which will be used to carry out link analysis and compute the structural similarity.

3.5 Link Analysis

Inspired by the underlying idea of link analysis algorithm—SimRank [11] that two objects are similar if they are related to similar objects, we propose a graph-based method to select the related keywords for short text enrichment. As in Figure 3, for a node w in the bipartite graph, we denote the set of its in-neighbors by $I(w)$, and individual in-neighbors are denoted as $I_i(w)$, for $1 \leq i \leq |I(w)|$. The SimRank score between node w_a and w_b is computed by

$$s(w_a, w_b) = \begin{cases} 1 & , \quad \text{if } w_a = w_b \\ \frac{C}{|I(w_a)||I(w_b)|} \sum_{i=1}^{|I(w_a)|} \sum_{j=1}^{|I(w_b)|} s(I_i(w_a), I_j(w_b)), & \text{if } w_a \neq w_b \end{cases}, \quad (3)$$

Where $C \in (0,1)$ is a decay factor. Specifically, the SimRank score is defined to be 0 when $|I(w_a)| = \emptyset$ or $|I(w_b)| = \emptyset$. According to (3), SimRank is symmetrical that $s(w_a, w_b) = s(w_b, w_a)$. Additionally, SimRank is an iterative fix-point algorithm, and its time complexity is $O(knd)$, where k is the number of iterations, n is the number of nodes, and d is the average of the in-degree of leaf nodes.

In our case, the weights on the edges in Figure 3 indicate the salient level of the corresponding keyword under the specific topic. However, the native SimRank algorithm fails to properly utilize the weights to enhance the possibility of selecting the

most representative keywords to enrich short text for classification. So we propose to use (6) to compute the topical SimRank score,

$$SR(w_a, w_b) = SAS(w_a)SAS(w_b)s(w_a, w_b), \quad (4)$$

In Figure 3, using the modified SimRank, we can obtain that w_2 is more similar to w_i than w_1 , because w_1 is shared by more than one topic and with low salient level. This merit ensures that we can expand short text and try our best to avoid introducing noise.

As Figure 4 shows that the keywords distribution under topics follow the long-tail distribution. Some keywords may be shared by many topics because they are relevant to all of the topics. Fortunately, the modified SimRank can be used to further purify these shared keywords.

3.6 Short Text Expansion

With the aim of resolving the sparsity problem in short text feature representation and avoiding noise, we propose to discover topic keywords to enrich the original short text. For classification task, the keywords shared by many topics are considered to be noise or with little discriminability. Thus, the most likely candidate keywords selected as expanding information are those that with few topics.

As in Figure 1, each leaf node is corresponding to a topic distribution, which is a column vector in the matrix $\Phi = \{\varphi_i\}$. For achieving high reliability, the K-L divergence of the topics distribution on the candidate keywords is integrated with SimRank score using Equation (1) to discern the most similar keywords.

In order to obtain $CScore(sw_i, cw_j) = CScore(cw_j, sw_i)$, we apply symmetrical-version of K-L divergence as in (5),

$$KL(sw_i, cw_j) = \frac{1}{2} [D(p_{sw_i}^{(z)} \parallel \frac{p_{sw_i}^{(z)} + p_{cw_j}^{(z)}}{2}) + D(p_{cw_j}^{(z)} \parallel \frac{p_{cw_j}^{(z)} + p_{sw_i}^{(z)}}{2})] \quad (5)$$

Where $D(p \parallel q) = \sum_k p_k \log \frac{p_k}{q_k}$.

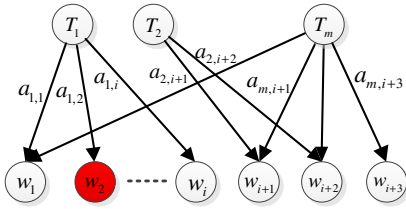


Fig. 3. Topic-keyword bipartite subgraph

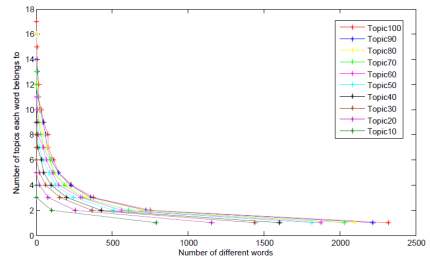


Fig. 4. The keywords distribution under topics

Finally, for each seed word, all the likely candidate keywords discovered from the topic-keyword graph are ranked in decreasing order according to Equation (1) and the top- ν candidates are selected for short text enrichment. Our proposed method is a graph-based model that exploits topics as background knowledge and synthesizes both semantic structure representation and similarity computation. Experimental results show that our method is effective and can outperform the state-of-the-art techniques.

4 Experiments

To validate the effectiveness of our proposed method, we conducted experiments on two real-world datasets: Search snippets and 20Newsgroup.

Search snippets dataset, collected by Phan X. H. [4], consists of 10,060 training snippets and 2,280 test snippets from 8 categories, as shown in Table 3. On average, each snippet has 18.07 words.

20Newsgroups is a standard corpus including 18,846 messages from 20 different Usenet newsgroups. Each newsgroup is corresponding to a different topic.

4.1 Evaluation on Search Snippets

Based on search snippets dataset, we choose MaxEnt and LibSVM as classifiers to evaluate the qualities of our methods for feature representation. Among various machine learning methods, MaxEnt and SVM have been successfully applied in many text mining tasks [18], which proves that MaxEnt is much faster in both training and inference while SVM is more robust. For comparisons, we employ LDA as the usage in [4] as baseline.

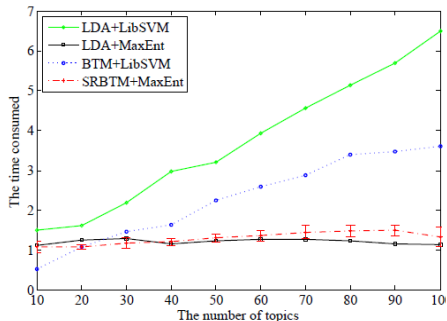
In baseline, we firstly learn LDA model based on external Wikipedia corpus as [4] to estimate its parameters. Then, the latent topics drawn by LDA are used to expand original short text. For evaluation, the topics distribution corresponding to each snippet is fed to LibSVM classifier, and the expanded snippets with latent topics are leveraged as new features for MaxEnt classifier. Meanwhile, BTM is trained directly on the search snippets, and our method SRBTM as shown in Figure 1 is applied to discover related keywords to enrich the short snippets.

In the proposed method SRBTM, we select the top- ν candidate keywords as enriching features when topic number is a constant k . In order to provide substantial results to prove the effectiveness of our method, we assign $\nu=[0,1,2,\dots,10]$ for a fixed topic number.

Then, we compute the average and variance of classification accuracy when ν changes, and the result are shown in Figure 6(a). We can find that our method outperforms the baselines obviously, and obtain the highest accuracy of 0.8678 when $k=10, \nu=9$, which reduce classification error by 10.01% compared to [3] and by 25.52% compared to [4].

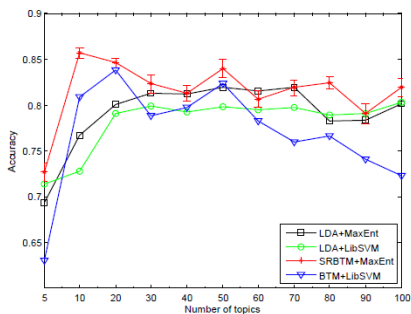
Table 3. Details of search snippets

Domain	Tr_snippets	Te_snippets
Business	1200	300
Computers	1200	300
Cult.-arts-ente.	1880	330
Edu.-Science	2360	300
Engineering	220	150
Health	880	300
Politics-Society	1200	300
Sports	1120	300
Total	10060	2280

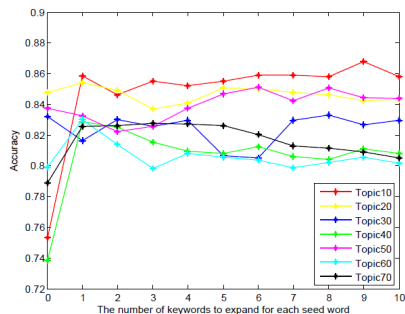
**Fig. 5.** Time consumed in test stage

When v varies from 0 to 10, the classification results of MaxEnt classifier are demonstrate in Figure 6(b). It is clear that enriching text representation using related keywords can indeed introduce useful information and achieve significant quality enhancement in the classification performance. However, when v is large enough, no more useful information can be added. Especially, when topic number $k=10$, we can obtain best result.

At last, the time consumed of classifiers over variational topics in prediction stage are compared in Figure 5. It is obviously that the MaxEnt classifier is faster and more stable than LibSVM. The LibSVM classifier will consume more time with the increase of the dimensionality of features. Our method proposed in this paper to enrich short text consume comparative time with that of LDA.



(a)



(b)

Fig. 6. Accuracy vary with topic numbers and expanded keyword

4.2 Evaluation on 20Newsgroups

In previous experiments, we have demonstrated the effectiveness of SRBTM on short texts. Although we propose SRBTM for enriching the representation of short text, there is no limitation for our method to be applied on normal text. Therefore, it is also interesting to see whether SRBTM can alleviate the Synonyms and Polysemy problem

in normal text, and further enhance the classification performance. For this purpose, we compared SRBTM with sparse topical coding (STC), proposed by Zhu and Xing [17]. Based on 20Newsgroups, SRBTM is applied to identify topic-focused keywords and expand the original features.

Table 4. Accuracy vary with topic numbers on 20NG

Method \ Topic num.	10	20	30	40	50
STC	0.70	0.757	0.789	0.809	0.817
SRBTM+Liblinear	0.8196 (± 0.0026)	0.8170 (± 0.0018)	0.8126 (± 0.0047)	0.8117 (± 0.0037)	0.8073 (± 0.0066)
Method \ Topic num.	60	70	80	90	100
STC	0.788	0.818	0.812	0.784	0.775
SRBTM+Liblinear	0.8090 (± 0.0054)	0.8082 (± 0.0022)	0.7467 (± 0.1895)	0.8054 (± 0.0038)	0.7995 (± 0.0090)

In our experiments, we employ the output of SRBTM to learn Liblinear classifier, and with STC as baseline. Then, the comparison results are given in Table 4. As the similar settings to the prior experiments, we compute the average and variance of accuracy when v changes from 0 to 10. The results indicate that our approach can introduce discrimination information to some extent on normal text. When $k=10$, we achieve the highest average accuracy, which is consistent to the validation on short text. To interpret this result, we can find the underlying foundation from Figure 4 that with the increase of k , the number of keywords shared by many topics is increasing, which hurt the quality of likely candidate keywords discovered by SRBTM.

5 Conclusion and Future Work

In this paper, we presented a novel method to enrich short text for classification based on topic-keyword graph and link analysis. The topics drawn from the original short and noisy texts are mapped as cliques to form the topic-keyword graph, which depict the semantic structure of the whole corpus. Then, the link analysis algorithm—SimRank is applied to compute similarities between nodes from the link structure perspective. Finally, K-L divergence is incorporated with the output of SimRank to reliably select candidate words to perform short text enrichment.

The main contributions of our work are: (1) we improve the TFIDF-like score in [10] and use it to re-rank the BTM-derived topic keywords to refine the topic definitions and improve the coherence, interpretability and ultimate usability of learned topics; (2) based on the re-ranked topic keywords, a semantic graph is constructed and a topic weighted SimRank method is proposed to measure the affinity among nodes;

(3) a novel comprehensive similarity measurement is proposed to select the most related keywords for short text expansion without the large-scale external corpus.

In the future, we will study techniques to fully exploit the topic-keywords graph to reduce noise, and to combine the vector representation of words [19] to further improve the short text classification performance.

References

1. Sun, A.: Short text classification using very few words. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1145–1146. ACM (2012)
2. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 841–842. ACM (2010)
3. Chen, M., Jin, X., Shen, D.: Short text classification improved by learning multi-granularity topics. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume, vol. 3, pp. 1776–1781. AAAI Press (2011)
4. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, pp. 91–100. ACM (2008)
5. Zhang, L., Li, C., Liu, J., Wang, H.: Graph-based text similarity measurement by exploiting Wikipedia as background knowledge. *World Academy of Science, Engineering and Technology* 59, 1548–1553 (2011)
6. Hu, X., Sun, N., Zhang, C., Chua, T.S.: Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 919–928. ACM (2009)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022 (2003)
8. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International Conference on World Wide Web, pp. 377–386. ACM (2006)
9. Yan, X.H., Guo, J.F., Lan, Y.Y., Cheng, X.Q.: A bitern topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1445–1456 (2013)
10. Song, Y., Pan, S., Liu, S., et al.: Topic and keyword re-ranking for LDA-based topic modeling. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1757–1760. ACM (2009)
11. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM (2002)
12. Antonellis, I., Molina, H.G., Chang, C.C.: Simrank++: query rewriting through link analysis of the click graph. *Proceedings of the VLDB Endowment* 1(1) (August 2008)
13. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *IJCAI* 7, 1606–1611 (2007)
14. Evgeniy, G., Markovitch, S.: Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In: *AAAI*, vol. 6, pp. 1301–1306 (2006)

15. Zhu, Y., Li, L., Luo, L.: Learning to classify short text with topic model and external knowledge. In: Wang, M. (ed.) KSEM 2013. LNCS, vol. 8041, pp. 493–503. Springer, Heidelberg (2013)
16. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (1999)
17. Zhu, J., Xing, E.P.: Sparse topical coding. arXiv preprint arXiv:1202.3778 (2012)
18. Berger Adam, L., Pietra, V.J.D., DellaPietra, S.A.: A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1), 39–71 (1996)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

Sentence-Length Informed Method for Active Learning Based Resource-Poor Statistical Machine Translation

Jinhua Du^{1,2}, Miaomiao Wang^{1,2}, and Meng Zhang¹

¹ School of Automation and Information Engineering, Xi'an University of Technology

² Shaanxi Key Laboratory of Complex System Control and Intelligent Information
Processing, Xi'an, 710048 China
jhdu@xaut.edu.cn

Abstract. This paper presents a simple but effective sentence-length informed method to select informative sentences for active learning (AL) based SMT. A length factor is introduced to penalize short sentences to balance the “**exploration**” and “**exploitation**” problem. The penalty is dynamically updated at each iteration of sentence selection by the ratio of the current candidate sentence length and the overall average sentence length of the monolingual corpus. Experimental results on NIST Chinese–English pair and WMT French–English pair show that the proposed sentence-length penalty based method performs best compared with the typical selection method and random selection strategy.

Keywords: active learning, SMT, sentence length penalty.

1 Introduction

The statistical-based or corpus-based machine translation (SMT) is intrinsically a kind of data-driven method, thus, the scale and quality of the parallel data are crucial for obtaining a good translation performance, especially the large-scale high quality parallel data.

However, it is not the case for many resource-poor language pairs. A number of methods have been presented to alleviate this problem, such as paraphrasing [1–3], related rich resources [4], etc. Considering the reality that a large scale of monolingual data can be easily acquired from the Web, digital media etc., active learning framework for SMT has been proposed to facilitate the shortage issue of parallel data [5–10]. Thus, less human cost could bring a significant improvement to the translation performance of resource limited language pairs.

The key issue in AL strategy is to choose rich-information sentences. As to the SMT, the basic idea of selecting sentences with high information is to find some sentences at each iteration to make the improvement of translation quality maximum [5]. In doing so, the sentences selected are of rich information. Intuitively, if more phrases or words in a sentence occur in the unlabeled (monolingual) data, then it might be more informative because it introduces more new

knowledge [5]. In this paper, we present a sentence length informed method to alleviate the tendency of choosing shorter sentences if the unlabeled data has a large variation in sentence length.

Using the state-of-the-art translation units based method and random selection method as baselines, our proposed method shows significant improvement in terms of translation quality compared with baselines on NIST Chinese-English pair and WMT French-English pair.

2 Related Work

In 2009, Haffari et al. (2009) firstly proposed a practical active learning framework for SMT where a number of high-quality parallel data are acquired from the large-scale monolingual data [5, 6]. Experimental results show that generally the translation unit based selection strategies, namely phrases and n -grams, performed best compared to other methods, such as random selection, translation confidence, inverse model etc.

In 2010, Ambati et al. proposed an active crowd translation (ACT) paradigm where active learning and crowd-sourcing come together to enable automatic translation for low-resource language pairs. Active learning is used to reduce cost of label acquisition by prioritizing the most informative data for annotation, while crowd-sourcing reduces cost by using the power of the crowds to meet up the lack of expensive language experts. Their experiments showed significant improvements in translation quality even with less data [7, 9].

In 2012, Bakhshaei and Khadivi applied a pool-based AL strategy to improve Farsi-English SMT system. They increased n in the n -gram feature from 4 to 5, and verified that the sentence selection algorithms such as translation units, translation confidence, inverse model etc. perform better than the random selection method in the task of Farsi-English translation [10].

On the basis of previous work, this paper introduced a length penalty factor into the phrase-based sentence selection strategy to penalize the short sentences. The penalty is dynamically updated at each iteration of sentence selection by the ratio of the current candidate sentence length and the overall average sentence length of the monolingual corpus.

3 Active Learning Framework for SMT

The AL framework for SMT is to obtain parallel data from the large-scale monolingual corpus and add to the initial small-scale parallel corpus for training.

We denote the initial parallel corpus as $L := \{(f_i, e_i)\}$, and the large-scale monolingual corpus as $U := \{f_j\}$. The key step is to design an algorithm to select highly informative sentences and submit to human translators.

Generally, the active learning framework has two prerequisites: (1) small-scale initial parallel corpus used to build a baseline SMT system; (2) large-scale monolingual corpus to acquire extra bilingual data. More importantly, there are two key issues in the AL strategy: (1) how to design an efficient algorithm to

evaluate the information that a sentence contains and select the rich-information sentences; (2) how to utilize the new parallel data to train and update the SMT system.

In our work, we mainly studied the first question, i.e., using translation units based methods, especially the phrase-based ones, on Chinese-English and French-English language pairs.

As to the second issue, different from the method used in [6], we only use the L trained model to run our SMT system at each iteration. Thus, the modified active learning framework using translation units based methods in our experiments is shown in “**Algorithm 1**”,

Algorithm 1. Modified AL-SMT

- 1: Given bilingual corpus L , and monolingual corpus U .
 - 2: $M_{F \rightarrow E} = \mathbf{train}(L)$
 - 3: **for** $t = 1, 2, \dots, N$ **do**
 - 4: Generate “Phrase Set” and compute sentence scores
 - 5: Select k sentences from U , and ask human experts
 for true translations.
 - 6: Remove the k sentences from U , and add the k
 sentence pairs to L .
 - 7: Update $M_{F \rightarrow E} = \mathbf{train}(L)$
 - 8: Evaluate the system performance on the test set.
 - 9: **end for**
-

4 Sentence-Length Informed Selection Strategy

The general mathematical description of a sentence selection algorithm is that given a monolingual corpus U , an initial parallel corpus L , and a sentence s consisting of m possible translation units $\{x | x \in X_s^m\}$ in U , the goal is to choose a sentence s with the highest score ϕ under a certain metric F as the most informative candidate. Therefore, the metric F to evaluate how much information a sentence has is most important in a selection algorithm. We can see that this process can be defined as a *quadruple* in (1),

$$\phi(s) = F(X, s, U, L) \quad (1)$$

4.1 Geom-Phrase and Arith-Phrase Algorithms

In these two methods, the basic unit for computing scores of a sentence is phrase. The Geom-Phrase algorithm is as in (2),

$$\phi(s) = \left[\prod_{x \in X_s^m} \frac{P(x|U)}{P(x|L)} \right]^{\frac{1}{|X_s^m|}} \quad (2)$$

where X_s^m is the set of possible phrases that the sentence s can offer, $P(x|U)$ and $P(x|L)$ is probabilities of observing x in U and L respectively, which are calculated as in (3) and (4),

$$P(x|U) = \frac{\text{count}(x) + \epsilon}{\sum_{x \in X_U^m} \text{count}(x) + \epsilon} \quad (3)$$

$$P(x|L) = \frac{\text{count}(x) + \epsilon}{\sum_{x \in X_L^m} \text{count}(x) + \epsilon} \quad (4)$$

where ϵ is the smooth factor ¹. X_U^m indicates the set of phrases that indeed occur in U , and X_L^m represents the set of phrases that truly appear in L .

The Arith-Phrase method is defined as in (5):

$$\phi(s) = \frac{1}{|X_s^m|} \sum_{x \in X_s^m} \frac{P(x|U)}{P(x|L)} \quad (5)$$

In [6], the phrases in Eq. (2) and Eq. (5) are extracted from the k -best list of translations of a sentence s in U . In addition, the out-of-vocabulary (OOV) words appeared in the translations are also included as candidate phrases with a uniform probability. In order to make the phrase set approximately as a complete set, that is, include all possible phrases that a sentence can offer, we utilize the phrase table generated by L to retrieve all possible phrases and collect OOVs that are not occurred in the phrase table.

4.2 Sentence-Length Informed Algorithm

The idea of presenting the sentence-length informed method is inspired by the findings and analysis in Section 5. The experimental results are not consistent with the conclusions in [5]. Thus, we carried out a comprehensive investigation and analysis, and found that

- the sentences selected by Arith-Phrase algorithm are generally shorter than the random selection;
- the sentence length varies in a wide range in corpus U ($1 \sim 100$ words in a sentence in our experiments).

We consider that the sentence length might have a significant impact on the selection performance. Therefore, we introduce a brevity penalty to prevent very short sentences as in [12]. The modified Arith-Phrase algorithm which we call it ‘‘Arith-Phrase-Penalty’’ is as in Eq. (6),

$$\phi(s) = \left[\frac{1}{|X_s^m|} \sum_{x \in X_s^m} \frac{P(x|U)}{P(x|L)} \right] \times BP \quad (6)$$

¹ We set $\epsilon = 0.5$ in the experiments.

where BP is the brevity penalty and defined as follows,

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases} \quad (7)$$

where r is the average sentence length in the monolingual corpus U , c is length of the sentence to be selected. Note that r is dynamically updated at each iteration with the change of the monolingual corpus U after the informative sentences are selected out.

5 Experiments, Findings and Analysis

5.1 Experiment Setup

The language pairs in our experiments are Chinese–English and French–English. English is the target language both in these two pairs. The initial parallel data and monolingual data are randomly selected respectively from NIST Chinese–English FBIS corpus and WMT News Commentary corpus, where the parallel data contains 5k pairs and the monolingual data includes 20k sentences.

The development set for Chinese–English task is NIST 2006 current set (1,664 sentences with four references for each source sentence), the test sets are NIST 2005 current set (1,083 sentences with four references for each source sentence) and 2008 current set (1,357 sentences with four references for each source sentence). The development set for French–English task is WMT Newstest 2013 (3,000 sentences with one reference for each source sentence), and the test set is WMT Newstest 2014 (3,003 sentences with one reference for each source sentence).

We utilize Moses [11] to indirectly evaluate the performance of sentence selection algorithms in terms of BLEU scores. The language model is five-gram built on the English part of the bilingual corpus.

As in [5], the iteration times in the AL framework is set to 25, and at each iteration, 200 informative sentences are chosen from the corpus U ; the smooth factor ϵ in Eq. (3) and Eq. (4) is set to 0.5.

5.2 Experiments on Arith-Phrase and Random Methods

We set the random selection method as the basic baseline. In this Section, we carried out a comparison experiments between the typical Arith-Phrase algorithm and the baseline on Chinese–English and French–English pairs, and found that the typical Arith-Phrase method did not beat the random selection method as shown shown in Figure 4.

In Figure 4, three top figures demonstrate the true BLEU scores at each iteration of the random, typical Arith-Phrase and our proposed methods for two language pairs, and three bottom figures use the polynomial fitting to demonstrate the trends of these methods so that the differences in terms of BLEU can be obviously observed.

We can see that in our experiments BLEU scores of the typical Arith-Phrase method are significantly lower than the random method.

A comprehensive investigation and analysis were carried out whereafter, and found that the sentences selected by the typical Arith-Phrase algorithm are generally shorter than those of the baseline. The observation drives us to consider the following questions,

- What extent does the sentence length affect the performance of algorithms?
- How could we alleviate the impact of sentence length so that we can find out the true informative sentences?

With these questions, we proceeded a data analysis as described below.

5.3 Data Statistics and Analysis

5.3.1 The Contradiction

Here we define “new words” as occurring in U but not in L before, and “existing words” as appearing both in U and L .

When selecting high-information sentences, there is a pair of contradiction that is **exploration** and **exploitation**, i.e. selecting sentences to discover new phrases vs estimating accurately the phrase translation probabilities [5]. Specifically,

- the more new words a sentence has in terms of the parallel corpus, the more informative the sentence is, but a lower word alignment accuracy to the added parallel data (c.f. Section 5.3.4);
- while the more existing words a sentence has to the parallel data, the more accurate the phrase probability is estimated, but a lower coverage to the test set (or unknown data)(c.f. Section 5.3.3 and Section 5.3.4).

Thus, we need to make a tradeoff between the ratio of new words and existing words when selecting out sentences from the monolingual corpus U .

Accordingly, regarding the negative results, we conducted a data analysis to investigate the hidden reasons from three aspects:

- statistics of the sentence length in different situations;
- the coverage rates of test sets by the parallel data in different situations;
- the increments of existing words and new words in different situations.

5.3.2 Average Sentence Length

Figure 1 illustrates the average sentence length of selected sentences at each iteration of the Random, Arith-Phrase and Arith-Phrase-Penalty methods for two language pairs.

In Fig. 1, “X” in “Random:X”, “Arith-Phrase:X” and “Arith-Phrase-Penalty:X” indicates the overall average sentence length of all selected sentences in terms of three different methods; “Average Sentence Length of U” is the average sentence length of the monolingual corpus U at each iteration that is in fact the parameter r in Eq. (7).

It can be seen that

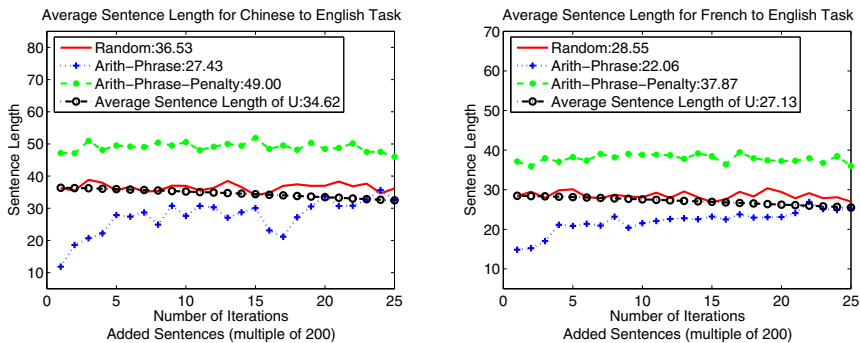


Fig. 1. Statistics of sentence length for different selection methods

- the average length of Arith-Phrase is shortest at each iteration both for two language pairs;
- the variance of the average sentence length of Arith-Phrase is biggest while the average lengths of the other two methods change slightly at each iteration.

The main purpose of active learning based SMT is to find and select the most informative sentences from the monolingual corpus, then the observations above drive us to ask that do the short sentences selected by Arith-Phrase algorithm really contain more information? If so, why does it perform worse than the Random method?

To answer the questions above, we might investigate the deep relationship between “existing words” and “new words” – the contradiction of the active learning SMT.

5.3.3 Coverage of Testsets by Parallel Data

An investigation is carried out to compute the distribution of the coverage rates of the test set by selected sentences at each iteration to see whether the Arith-Phrase method can find out more informative sentences. Results are shown in Figure 2².

We can see that the coverage of Arith-Phrase is significantly higher than that of Random. This indicates that Arith-Phrase tends to select sentences containing more “new words”, i.e., tends more to the “exploration” side. Intuitively, the increase of new words would improve the translation performance because it is useful to reduce the out-of-vocabulary (OOVs) in the translation hypotheses. However, higher coverage rate did not improve the translation quality!

We analyze that this might be: the Arith-Phrase algorithm is potentially to look for highly informative sentences featured by more new words. Intuitively, the shorter a sentence is, the greater the proportion of news words in this sentence is, the more likely it can be chosen compared to a longer sentence.

² Due to the limitation of the paper space, we take ZH-EN NIST 2005 and FR-EN WMT 2014 test sets as examples. It is the same situation for ZH-EN NIST 2008.

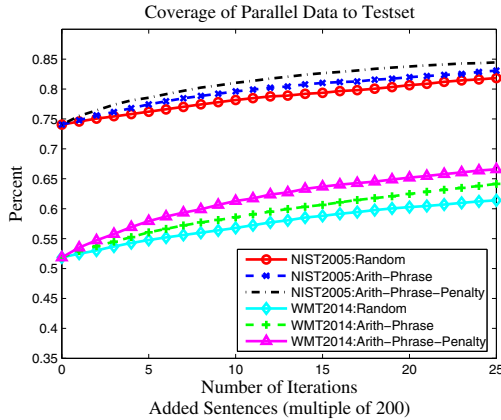


Fig. 2. Comparison of coverage rates of parallel data to different test sets at each iteration

The possible reason is that although Arith-Phrase tries to make a tradeoff between the “**exploitation**” and “**exploration**”, it is difficult when the sentence length varies in a wide range of the monolingual corpus. From another viewpoint, the increase of new words implies that the relative decrease of frequencies of existing words that might be more important to improve the accuracy of phrase probability estimation for the resource-limited small-scale SMT system.

Based on the data statistics and analysis of sentence length and coverage, we refer that the variation between the frequencies of existing words and new words might provide a reasonable explanation.

5.3.4 Statistics of Existing Words and New Words

As mentioned that “existing words” appear both in U and L , and “new words” occur only in U , we analyze the relationship between existing words and new words from three aspects: 1) the incremental frequencies of existing words at each iteration; 2) the incremental new words at each iteration; 3) the incremental frequencies of new words at each iteration. Statistics are shown in Fig. 3.

It can be seen in Fig.3 that

- in terms of incremental frequencies of existing words for two language pairs, the Random method is significantly higher than the Arith-Phrase algorithm. This would improve the accuracy of probability estimation of existing words.
- in terms of incremental new words for two language pairs, the Arith-Phrase is higher than the Random method. This is consistent with the coverage comparison in Section 5.3.3.
- in terms of incremental frequencies of new words, the Arith-Phrase is also higher than the Random method. However, we can find that the frequency increments of new words are nearly the same as the increments of new words, i.e., the new words have quite low frequencies. For example, at iteration 5 of

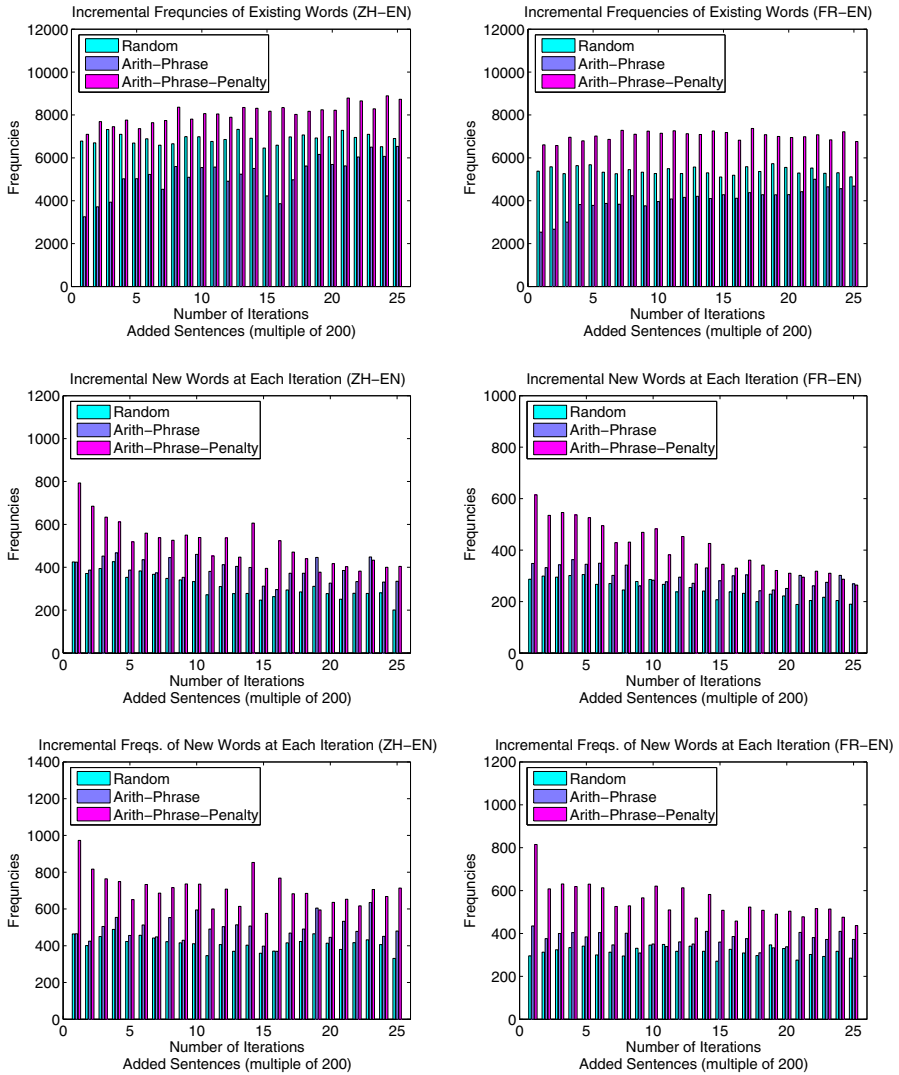


Fig. 3. Statistics of existing words and new words at each iteration

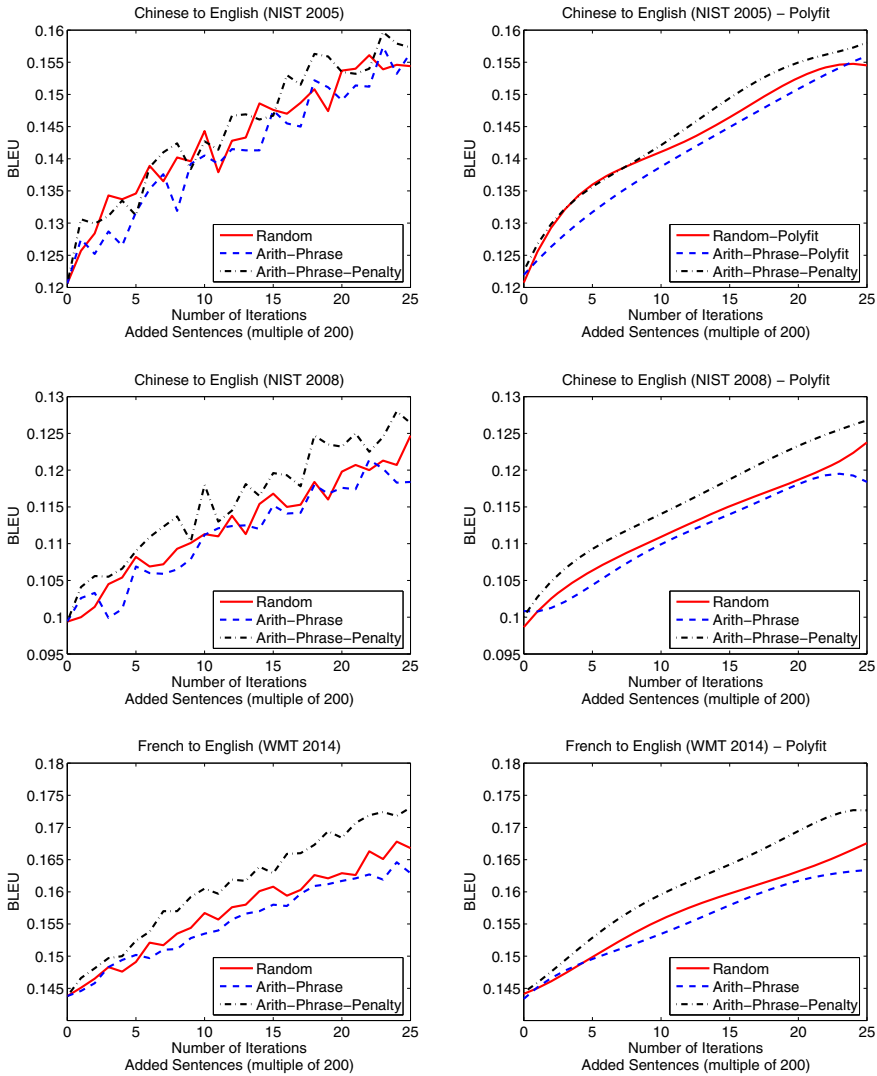


Fig. 4. Experimental results of Arith-Phrase, Sentence-length Informed (Arith-Phrase-Penalty) methods compared to the baseline (Random)

Arith-Phrase for FR-EN task, the number of incremental new words is 345, the number of incremental frequencies is 384, the average frequency for each new word is about 1.11 that is quite low. Thus, this might explain that for Arith-Phrase method, although the coverage increases, the accuracy of the probability estimation of existing words and new words relatively decreases due to low frequencies compared to the Random method.

Based on the analysis, we consider that if a sentence length informed factor can be introduced into the Arith-Phrase algorithm, it might balance the contradiction of existing words and new words. Thus, the brevity penalty based Arith-Phrase algorithm is proposed. The next sections will carry out comparison experiments between our method and baselines.

5.3.5 Experiments on the Proposed Method

It can be found in Figures 1, 2, 3 and 4 that

- the proposed Arith-Phrase-Penalty method significantly outperforms the Random and Arith-Phrase in all tasks in terms of translation performance (BLEU scores)(See Fig. 4);
- the incremental frequencies of existing words of Arith-Phrase-Penalty is far higher than those of Random and Arith-Phrase, which could further improve the probability estimation accuracy of existing words (See Fig. 3);
- the incremental new words of Arith-Phrase-Penalty is higher than Random and Arith-Phrase, which could bring a broader coverage to test sets. Figure 2 shows the consistency;
- the average sentence length of Arith-Phrase-Penalty is larger than Random and Arith-Phrase and the variance is small (See Fig. 1). The longer sentences bring more existing words and introduce new words into the parallel data, which not only increases the coverage to test sets, but also improves the accuracy of probability estimation of phrases. However, the potential problem is that the human cost would be risen.

It can be said that the proposed algorithm indeed improved the performance of typical Arith-Phrase method, and achieved best results.

6 Conclusions

This paper studies the active learning framework and different high-information sentence selection algorithms for resource-poor SMT. Based on the negative experimental results on Arith-Phrase method, we found that the sentence length is an important factor to affect the system performance when the length of sentences in the monolingual corpus varies in a wide range. Based on the analysis, a simple but effective method – sentence length informed Arith-Phrase – is proposed to penalize sentences that are shorter than the overall average length of the monolingual corpus U at each iteration. Experimental results demonstrate

that the proposed method significantly outperforms the typical Arith-Phrase and Random method.

In future, we intend to carry out further study on the AL framework in the respects of 1) presenting improved sentence selection algorithms that contain rich knowledge to better quantize the information in a sentence; 2) proposing novel solutions that can better balance the tradeoff between **exploration** and **exploitation**, and decrease the human cost in the AL framework.

Acknowledgments. This work is supported by NSF project (61100085), the Open Projects Program of National Laboratory of Pattern Recognition, and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry. Thanks the reviewers for their insightful comments and suggestions.

References

- [1] Callison-Burch, C., Koehn, P., Osborne, M.: Improved statistical machine translation using paraphrases. In: Proceedings of HLT-NAACL 2006: Proceedings of the NAACL, pp. 17–24 (2006)
- [2] Nakov, P.: Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In: Proceedings of WMT, pp. 147–150 (2008)
- [3] Du, J., Jiang, J., Way, A.: Facilitating Translation Using Source Language Paraphrase Lattices. In: Proceedings of EMNLP, pp. 420–429 (2010)
- [4] Nakov, P., Ng, H.: Improved statistical machine translation for resource-poor languages using related resource-rich languages. In: Proceedings of EMNLP, pp. 1358–1367 (2009)
- [5] Haffari, G., Roy, M., Sarkar, A.: Active learning for statistical phrase-based machine translation. In: Proceedings of NAACL, pp. 415–423 (2009)
- [6] Haffari, G., Sarkar, A.: Active Learning for Multilingual Statistical Machine Translation. In: Proceedings of ACL and the 4th IJCNLP, pp. 181–189 (2009)
- [7] Ambati, V., Vogel, S., Carbonell, J.: Active learning and crowd-sourcing for machine translation. In: Proceedings of LREC, pp. 2169–2174 (2010)
- [8] Ambati, V., Vogel, S., Carbonell, J.: Multi-strategy approaches to active learning for smt. In: Proceedings of the MT Summit XIII, pp. 122–129 (2011)
- [9] Ambati, V., Hewavitharana, S., Vogel, S., Carbonell, J.: Active learning with multiple annotations for comparable data classification task. In: Proceedings of the Fourth Workshop on Building and Using Comparable Corpora, pp. 69–77 (2011)
- [10] Bakhshaei, S., Khadivi, S.: A Pool-based Active Learning Method for Improving Farsi-English MT system. In: Proceedings of IST, pp. 822–826 (2012)
- [11] Koehn, P., Hoang, H., Callison-Burch, C., et al.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of ACL, pp. 177–180 (2007)
- [12] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of ACL, pp. 311–318 (2002)

Detection of Loan Words in Uyghur Texts

Chenggang Mi^{1,2}, Yating Yang¹, Lei Wang¹, Xiao Li¹, and Kamali Dalielihan¹

¹ Xinjiang Technical Institute of Physics & Chemistry of Chinese Academy of Sciences,
Urumqi, Xinjiang 830011, China

² University of Chinese Academy of Sciences, Beijing, 100049, China
michenggang@gmail.com,
{yangyt, wanglei, xiaoli}@ms.xjb.ac.cn,
kamaly330@gmail.com

Abstract. For low-resource languages like Uyghur, data sparseness is always a serious problem in related information processing, especially in some tasks based on parallel texts. To enrich bilingual resources, we detect Chinese and Russian loan words from Uyghur texts according to phonetic similarities between a loan word and its corresponding donor language word. In this paper, we propose a novel approach based on perceptron model to discover loan words from Uyghur texts, which consider the detection of loan words in Uyghur as a classification procedure. The experimental results show that our method is capable of detecting the Chinese and Russian loan words in Uyghur Texts effectively.

Keywords: loan words detection, phonetic similarity, Uyghur, perceptron-based model.

1 Introduction

Statistical methods are commonly used in recent NLP tasks [1], which rely on corpora heavily. For tasks like SMT (Statistical Machine Translation), there always exist data sparseness during training of translation models because lack of bilingual texts [2]. This situation may get worse in under-resources languages' (like Uyghur) processing.

We find that there are many loan words in Uyghur, which are mainly borrowed from Chinese and Russian (Table 1), and a loan word always pronounces similarly with its corresponding donor language word. This might be an interesting clue to discover loan words in Uyghur texts. And further enrich Uyghur related bilingual resources.

To detection loan words in Uyghur, we consider it as a string similarity problem and transform phonetic similarity into string similarity firstly. Intuition might suggest that common used string similarity algorithms can solve this problem easily. However, spelling of loan words may change when borrowed from the donor language, also, the characters asymmetrical transformation affect the performance of detection model. Additionally, Uyghur words are forming as adding suffixes after a certain word stem, how to deal with these suffixes properly when compute string similarity also should be considered carefully. In this paper, we suggest a novel method to detect Chinese and Russian loan words in Uyghur texts, which can be described as following two parts: Characters alignment and Detection of Chinese and Russian loan words in Uyghur texts.

The rest of this paper is organized as follows. Section 2 describes previous work on loan words research. In section 3, we give an overview of loan words in Uyghur. The character alignment model and loan words detection approach are presented in section 4. Section 5 illustrates the corpus and gives the experimental results. Conclusions and future work are summarized in section 6.

2 Related Work

Previous works on loan words are mainly focused by linguists. [3] looks specifically at the language contact of English and Chinese and details the resultant language change when words from International English are borrowed into standard Chinese; [4] outlines the historical and cultural contexts of borrowing from English into Japanese, processes of nativization, and functions served by English loan words; [5] studied Chinese loan words in English; [6] concerned about loan words in English and Chinese, and the characteristics of their language contact. For loan words in Uyghur, [7] compared different methods that words borrowed by Uyghur and Chinese; [8] analyzed influence to Uyghur words made by loan words; [9] explained the historical process of the Chinese words borrowed into the Uyghur language, the characteristics of the Chinese borrowed words, their development and certain troubles of them in the course practical usage. In NLP field, some related works also proposed by researchers. [10] presented a string similarity based method to discover Chinese loan words in Uyghur, which combine two basic string similarity algorithms as a recognize model.

In this paper, we propose a novel method to detect Chinese and Russian loan words from Uyghur texts, which extend previous work, and consider the detection as a binary classification based on perceptron model. For minimum differences between donor languages (Chinese and Russian) and Uyghur, we obtain character mapping rules by characters aligning; features used during the perceptron model' s training are taken from five string similarity algorithms.

3 Loan Words in Uyghur

3.1 Introduction of Loan Words

A loanword is a word borrowed from a donor language and incorporated into a recipient language directly, without translation. Loan words may have several changes when loaned: 1) Changes in meaning; 2) Changes in spelling; 3) Changes in pronunciation.

3.2 Loan Words in Uyghur

Uyghur is an official language of the Xinjiang Uyghur Autonomous Region, In addition to influence of other Turkic languages, Uyghur has historically been influenced strongly by Persian and Arabic, and more recently by Mandarin Chinese and Russian.

Except named entity words like names of person and locations, there are also many regular terms borrowed from Chinese and Russian. We give some examples of loan words in Table 1:

Table 1. Examples of loan words in Uyghur

Chinese loan words in Uyghur [in English]		Russian loan words in Uyghur [in English]	
شىنجاڭ (新疆)	[Xinjiang]	رومكا (рюмка)	[cup]
لەنگەمن (拉面)	[noodles]	تېلېفون (телефон)	[telephone]
لازا (辣子)	[hot pepper]	ئۇنىۋېرسىتېت (университет)	[university]
شۇجى (书记)	[secretary]	رادىيو (радио)	[radio]
كوي (块)	[Yuan]	پوچتا (почта)	[post office]
لەنگېۋەك (凉粉)	[agar-agar jelly]	ۋېلسىپېت (велосипед)	[bicycle]
دۇفۇ (豆腐)	[bean curd]	ئوبلاست (область)	[region]

3.3 Challenges in Loan Words Detection

Challenge One: Spelling change when borrowed from donor languages (Chinese and Russian). The word of Uyghur and Russian can be writing as Latin alphabet, the Chinese word can be presented by Pinyin, for example:

Russian loan words in Uyghur: “رادىيو” (“radyo”) - “радио” (“radio”)

Chinese loan words in Uyghur: “كوي” (“koi”) - “块” (“kuai”)

Changes of spelling have a great impact on the loan words detection task.

Challenge Two: Suffixes of Uyghur words effect the detection of loan words.

A Uyghur word is composed of a word stem and several suffixes, which can be formally described as:

$$Word = stem + suffix_0 + suffix_1 + \dots + suffix_N \tag{1}$$

If we use the Edit Distance to measure the string similarity between a word and its original form, the length of the word’s suffixes equal even greater than the original word’ length. Even though the word is a loan word actually, traditional similarity algorithms cannot give a sure result.

For overcome above two challenges, we propose a loan words detection approach, which can be divided into two steps: 1) Characters Alignment (in section 4.1, for overcome Challenge One); 2) Classification-based loan words detection model (in section 4.2, for overcome Challenge Two).

4 Recognition of Loan Words from Uyghur Texts

In this paper, we propose a novel method to discover loan words from Uyghur texts, which combines several string similarity algorithms as feature functions of a perceptron-based model, and consider the loan words detection in Uyghur texts as a binary classification problem. Rather than transforming Uyghur characters according to traditional rules, associations between Uyghur characters and donor language (Chinese(pinyin) and Russian) characters are obtained by aligning exist words (denote as character sequences).

4.1 Characters Alignment

To obtain characters mapping rules, we consider it as a word alignment problem, which take donor language characters as source language words, recipient language characters as target language words. The IBM Model 1 [11] and HMM [12] are used here as word alignment models, parameters of these models are estimated by EM (Expectation Maximum) algorithm [13].

Two alignment models can be formally described as follows:

IBM Model 1

$$p(e, a|f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}), 1 \leq j \leq l_e. \tag{2}$$

l_f and l_e are the length of donor language characters (Chinese or Russian) and Uyghur characters; a is the word alignment function, $a: j \rightarrow i$ means source word f_i is align with target word e_j ; $t(e|f)$ is the translation probability of source word f and target word e . When training the IBM model 1, $t(e|f)$ can be computed based on word co-occurrence counts:

HMM

$$p(f|e) = \sum_{a_1^m} \prod_{j=1}^m [p(a_j|a_{j-1}, l) \cdot p(f_j|e_{a_j})] \tag{3}$$

Here, alignment probabilities are independent of word position and depend on jump width ($a_j - a_{j-1}$).

Suppose we have one pair of Uyghur-Chinese words, **شىنجاڭ** and “Xinjiang” (which is the Pinyin of Chinese word “新疆”). For convenience, in this paper, we denote Chinese words with Pinyin), the associations of characters can be listed as follows:



Additionally, for characters align to null, we assign them target characters according to transform rules (Latin rules for Uyghur).

4.2 Classification-Based Loan Words Detection Model

Features of String Similarity Algorithms

In this part, we investigate features of five string similarity algorithms, and extend some of them (Edit Distance and Common String) to adapt the tasks that detect loan words in Uyghur texts.

Position-Related Edit Distance

Because distinctions of words forming between Uyghur and donor languages, we cannot use the Edit Distance to measure the similarity between a donor language word (Chinese, Russian) and a Uyghur word, directly. Suppose we have a Chinese word “兰州” (“lanzhou”) and a Uyghur word “لانتجولارخا”, due to the suffixes of the Uyghur word, the number of deletions according to the Edit Distance algorithm equal (this example) even greater than the length of Chinese word (“lanzhou”), so the Uyghur word cannot be recognized as a Chinese loan word (which actually is).

Intuition might suggest that the stemming of a Uyghur word firstly can avoid such problems, however, this approach depend on performance of the Uyghur stemmer heavily. In this part, we propose a position-related edit distance (PRED) method, which tracing the procedure of edit distance computing, if continues deletion occurred at the end of words, these deletion number will be subtracted from the edit distance results. Experimental results show that our method (PRED) outperform the stem-based method and the basic edit distance algorithm.

$$PRED_{a,b}(i,j) = \begin{cases} ED_{a,b}(i,j) & \text{No Continue Delete Occurred,} \\ ED_{a,b}(i,j) - times_{delete}(a,b) & \text{Otherwise.} \end{cases} \quad (4)$$

$times_{delete}(a,b)$ is the times continue deletion occurred at end of words.

Dice Coefficient

The coefficient may be calculated for two strings, a and b using bigrams as follows:

$$DC_{a,b} = \frac{2n_t}{n_a + n_b} \quad (5)$$

Where n_t is the number of character bigrams is found in both strings, n_a is the number of bigrams in string a and n_b is the number of bigrams in string b .

Weighted Common Subsequence

Rather than using the Longest Common Subsequence, we present a Weighted Common Subsequence (WCS) to measure similarity between two words, which assign a weight to each common subsequence according to its length. Finally, we sum up these results as WCS of two words (a and b).

$$WCS_{a,b} = \sum_{i=2}^{\min(L_a, L_b)} LEN_i \cdot NUM_i \quad (6)$$

L_a and L_b are length of word a and word b , respectively. LEN_i is the length of the i th common string, NUM_i is the number of these common strings appeared.

Jaccard Similarity Coefficient and *Overlap Similarity* ($JSC_{a,b}$, $OLS_{a,b}$) are also used as basic features.

Accordingly, Jaccard Similarity Coefficient, Overlap Similarity and Dice Coefficient are mainly focus on discrete string similarity; Position-Related Edit Distance can measure global similarity of two strings, which also overcome the shortage of basic Edit Distance in loan words detection task. The Weighted Common String algorithm measures string similarity of two words according to number of common strings and length of common strings. For detect loan words in Uyghur texts effectively, we consider these five string similarity algorithms as five feature functions of a binary classification model.

Perceptron-Based Loan Words Detection

The perceptron [14, 15, 16] is an algorithm for learning a binary classifier: a function that maps its input \mathbf{x} (which is a real-value vector) to an output value $f(\mathbf{x})$ (which is a single binary value)

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Where \mathbf{w} is a vector of real-valued weights, $\mathbf{w} \cdot \mathbf{x}$ is the dot product (which here computes a weighted sum), and b is the ‘‘bias’’, which is a constant term that does not depend on any input value. The value of $f(\mathbf{x})$ (0 or 1) is used to classify \mathbf{x} as either a positive or a negative instance, in the case of a binary classification problem. If b is negative, then the weighted combination of inputs must produce a positive value greater than $|b|$ in order to push the classifier neuron over the 0 threshold. Spatially, the bias alters the position of the decision boundary. The perceptron learning algorithm does not terminate if the learning set is not linearly separable. If the vectors are not linearly separable will never reach a point where all vectors are classified properly.

In this paper, we consider detection of loan words in Uyghur texts as a perceptron-based classification problem. Here, \mathbf{x} is a real-value vector which contains string similarities of two words, and compute by algorithms described in 4.2. The output of this model is a loan word label (0: no, 1: yes).

The vector \mathbf{x} used in the loan words detection task can be formally described as follows:

$$\langle PRED_{a,b}, DC_{a,b}, WCS_{a,b}, JSC_{a,b}, OLS_{a,b} \rangle \quad (8)$$

$PRED_{a,b}$, $DC_{a,b}$, $WCS_{a,b}$, $JSC_{a,b}$ and $OLS_{a,b}$ are string similarity scores of two words (a donor language word and a Uyghur word). The computation methods of these scores are presented in section 4.2.

5 Experiments

In this section, we evaluate our method by detect Chinese and Russian loan words in Uyghur texts.

5.1 Set Up

Character transformation rules are obtained by GIZA++¹, which is widely used in word alignment, and implemented IBM models and HMM. We implement five string similarity algorithms, respectively. The classification model we use here is **XPerceptron**, which is a C++ implementation of the perceptron model.

Results of loan words detection are evaluated by R (Recall), P (Precision) and F1 (F-measure), respectively. A indicates a set of loan words output by our method; B is a set of words also output by our method but none of them is loan word and C is a set of loan words included in test set but did not output. Therefore, we can compute F1 as:

$$R = \frac{A}{A + C}, \quad P = \frac{A}{A + B}, \quad F1 = \frac{2 * P * R}{P + R} \quad (9)$$

5.2 Introduction of Corpora

In this paper, we use a Uyghur-Chinese city names mapping table and Uyghur-Russian city names table to train the transformation rules (as described in Section 4.1). The test sets of loan words detection are selected from web, which include several domains, such as government documents, news, daily life, etc.

5.3 Experiments

We calculate similarity features of donor language words and Uyghur words according to five string similarity algorithms. Then, features of each word pair and its loan word label will be considered as an input of perceptron-based detection model. For validate the effectiveness of our method, we also conduct experiments on stem-based and traditional transformation rules-based approaches. Results of these experiments are shown in Table 2, Table 3, Table 4 and Table 5, respectively.

5.4 Results and Analysis

Table 2 performs results of five basic string similarity algorithms (ED (Edit Distance), DC (Dice Coefficient), CS (Common String), OS (Overlap Similarity) and JSC (Jaccard Similarity Coefficient)), which are based on words. Experiments show that ED and CS outperform other three algorithms, that because DC, OS and JSC mainly focus on discrete characters, ED and CS concern much on characters that continuous.

Results of stem-based methods (Table 3) are slightly better than word-based (in Table 2), one possible reason is that Uyghur stemmer can overcome some situation that caused by suffixes when computing string similarity.

¹ <https://code.google.com/p/giza-pp/>

Table 2. Results of five basic algorithms (word-based)

	Chinese Loan Words			Russian Loan Words		
	R	P	F1	R	P	F1
ED	71.20	62.41	66.52	73.29	67.32	70.18
DC	69.22	60.98	64.84	70.02	62.41	66.00
CS	73.12	61.16	66.61	76.08	68.43	72.05
OS	69.25	60.73	64.71	70.29	63.78	66.88
JSC	69.10	61.98	65.35	71.81	64.59	68.01

Table 3. Results of five basic algorithms (stem-based)

	Chinese Loan Words			Russian Loan Words		
	R	P	F1	R	P	F1
ED	71.38	62.71	66.76	74.32	68.50	71.29
DC	69.30	61.92	65.40	71.43	63.09	67.00
CS	73.15	63.20	67.81	76.13	69.27	72.54
OS	70.02	60.94	65.17	70.82	64.50	67.51
JSC	69.83	62.51	65.97	72.61	65.08	68.64

Table 4. Results of five algorithms (two improved, word-based)

	Chinese Loan Words			Russian Loan Words		
	R	P	F1	R	P	F1
PRED	75.72	64.73	69.80	75.39	70.02	72.61
DC	69.78	62.33	66.35	71.64	63.25	67.18
WCS	74.39	64.36	69.01	78.01	72.34	75.07
OS	71.29	61.72	66.16	71.05	65.20	68.00
JSC	71.32	63.65	67.27	72.89	65.37	68.92

Table 5. Result of perceptron-based detection model

	Chinese Loan Words			Russian Loan Words		
	R	P	F1	R	P	F1
PBDM	78.82	68.30	73.18	81.03	73.22	76.93

In Table 4, ED and CS in five basic string similarity algorithms are improved as PRED (Position-Related Edit Distance) and WCS (Weighted Common String), respectively. Although word-based, performances of PRED and WCS outperform ED and CS both in Table 2 and Table 3, significantly. Besides, results of Table 4 are based on transformation rules obtained by character aligning, which also contribute to the performance of string similarity algorithms.

Table 5 performs results of perceptron-based detection model (PBDM), which is the integration of our method. The PBDM combines five string similarity algorithms (PRED (Position-Related Edit Distance), DC (Dice Coefficient), WCS (Weighted Common String), OS (Overlap Similarity) and JSC (Jaccard Similarity Coefficient)) as five features, and loan word labels as outputs, according to features of the perceptron. In our experiments, PBDM achieved the best performance. The most important reason is that, perceptron-based model integrate advantages of five string similarity algorithms, and the error-driven model much adaptive to our task. Interestingly, the performance of Russian loan words detection is outperforming Chinese loan words detection in all experiments, which may because the spelling method of Russian loan words is much closer with Uyghur.

6 Conclusion and Future Work

To detection loan words in Uyghur texts effectively, we transfer the phonetic similarity between donor language (Chinese and Russian in our paper) words and Uyghur words to strings similarity, and consider the detecting procedure as a classification problem. Experimental results show that with our method Chinese and Russian loan words can be recognized efficiently. In future work, we will focus on extraction of bilingual resources based on loan words, and extend this approach to other languages.

Acknowledgements. This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No.XDA06030400), West Light Foundation of Chinese Academy of Sciences (Grant No.LHXZ201301 XBBS201216), the Xinjiang High-Tech Industrialization Project (Grant No. 201412101) and Young Creative Sci-Tech Talents Cultivation Project of Xinjiang Uyghur Autonomous Region (Grant No. 2013731021).

References

1. Chris, M., Hinrich, S.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
2. Chung, C., Ho, C., Ping, C.: Using Sublexical Translations to Handle the OOV Problem in Machine Translation. *ACM Transactions on Asian Language Information Processing* 10(3), 1–20 (2011)
3. Lauren, A.H.L.: English Loanwords in Mandarin Chinese. The University of Arizona, Arizona (2002)
4. Gillian, K.: English loanwords in Japanese. *World Englishes* 14(1), 67–76 (1995)
5. Kui, Z.: On Chinese-English Language Contact through Loanwords. *English Language and Literature Studies* 1(2), 100–105 (2011)
6. Xuan, L., Lanqin, Z.: On Chinese Loanwords in English. *Theory and Practice in Language Studies* 1(12), 1816–1819 (2011)
7. Yan, C., Ping, C.: A Comparison on the methods of Uyghur and Chinese Loan Words. *Journal of Kashgar Teachers College* 32(2), 51–55 (2011)
8. Yan, Z.: Influence of Loan Words on the Words of Uygur Language. *Journal of Hubei University of Education* 28(1), 37–39 (2011)

9. Shiming, C.: New Research on Chinese Loan Words in the Uyghur Language. *N.W. Journal of Ethnology* 28(1), 176–180 (2011)
10. Mi, C., Yang, Y., Zhou, X., Li, X., Yang, M.: Recognition of Chinese Loan Words in Uyghur Based on String Similarity. *Journal of Chinese Information Processing* 27(5), 173–178 (2013)
11. Brown, P.E., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2), 263–311 (1993)
12. Vogel, S., Ney, H., Tillmann, C.: HMM-based word alignment in statistical translation. In: *Proceedings of the 16th Conference on Computational Linguistics*, pp. 836–841. Association for Computational Linguistics (1996)
13. Dempster, A., Laird, N., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* (39), 1–38 (1977)
14. Gallant, S.I.: Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks* 1(2), 179–191 (1990)
15. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: *Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing*, vol. 10. Association for Computational Linguistics (2002)
16. Dasgupta, S., Kalai, A.T., Monteleoni, C.: Analysis of perceptron-based active learning. *Learning Theory*, 249–263 (2005)

A Novel Rule Refinement Method for SMT through Simulated Post-Editing

Sitong Yang^{1,2}, Heng Yu^{1,*}, and Qun Liu^{1,3}

¹ Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences

² University of Chinese Academy of Sciences

³ CNGL, School of Computing, Dublin City University
{yangsitong, yuheng, liuqun}@ict.ac.cn

Abstract. Post-editing has been successfully applied to correct the output of MT systems to generate better translation, but as a downstream task its positive feedback to MT has not been well studied. In this paper, we present a novel rule refinement method which uses Simulated Post-Editing (SiPE) to capture the errors made by the MT systems and generates refined translation rules. Our method is system-independent and doesn't entail any additional resources. Experimental results on large-scale data show a significant improvement over both phrase-based and syntax-based baselines.

1 Introduction

The quality of Statistical Machine Translation (SMT) is generally considered insufficient for use without a significant amount of human correction [1]. In the translation world, the term post-editing often refers to the process of manually correcting SMT output. While there does exist some documented cases of success, post-editing for SMT systems has not really become mainstream among professional translators, the main concerns are: unlike humans, the translation systems fail to learn from the post-editors corrections and keep making the same kind of mistakes.

One possible solution to this problem is by automatic post-editing [1–4]. Most of these works use another SMT system to capture the repetitive errors of the original system [2, 3], training a monolingual system to translate the original result into a better one. This method could arguably reduce the number of common errors and produce better results, but at the same time, it brings two side-effects: first, the flexibility: the training of the post-editing SMT system takes a long time, making it hard to adapt to different translation scenarios. Second, the pipeline of SMT systems involves several intermediate parts like alignment and rule-extraction, which will introduce additional errors and degrade the overall performance of the system.

* Corresponding author.

Another promising direction is to utilize post-editing results to capture the errors made by the SMT systems, and use a supervised error-driven paradigm to reinforce the original system. This method can make SMT systems more adaptive to all kinds of translation scenario without increasing the complexity of the system [5]. However, this task is challenging in two regards. First, the training data is expensive to generate, since it needs massive manual work for post-editing, Second, due to the poor quality of SMT output, it is hard to clearly identify the error and make the right correction.

In this paper, we follow the second direction and present a novel error-driven rule refinement method for SMT. First, we use a simulated post-editing paradigm in which either non-post-edited reference translation or manually post-edited translation from a similar MT system are used in lieu of human post-editors (Section 2). This paradigm allows us to efficiently collect the training data without expensive manual work and also enable the system to function in real-time post-editing scenarios without modification. Then we calculate the editing distance [6] between the translation output and the reference to capture the translation errors (Section 3.1), then generate refined rules based on the edit operations (Section 3.2). Finally, to ensure the goodness of the generated rules, we introduce a simple and effectively heuristic algorithm for rule-filtration (Section 3.3). We apply our method to both phrase-based and syntax-based SMT systems and gain an overall improvement of 1.4 BLEU point without using any additional resources. We also carry out experiment on multiple domains and find that our method works well on both news and medical domains (Section 4).

2 Simulated Post-Editing

In post-editing scenarios, humans continuously edit machine translation outputs into high quality translations, providing an additional, constant stream of data absent in batch translation. The data consists of highly domain relevant reference translations that are minimally different from MT outputs, making them ideal for learning. However, true post-editing data is infeasible to collect during system development and internal testing, as standard MT pipelines require tens of thousands of sentences to be translated with low latency. To address this problem, [8] formulated the task of simulated post-editing, wherein pre-generated reference translations are used as a stand-in for actual post-editing. This approximation is equivalent to the case where humans edit each translation hypothesis to be identical to the reference rather than simply correcting the MT output to be grammatical and meaning-equivalent to the source.

Our work uses this approximation for building large scale training set for refined rule-extraction. In our simulated post-editing task, we first use a baseline SMT system to translate all sentences in the bilingual corpus, then use the target side of bilingual corpus as the approximate post-editing results of the output from the SMT system. In this way, we are able to capture translation errors without any additional resource.

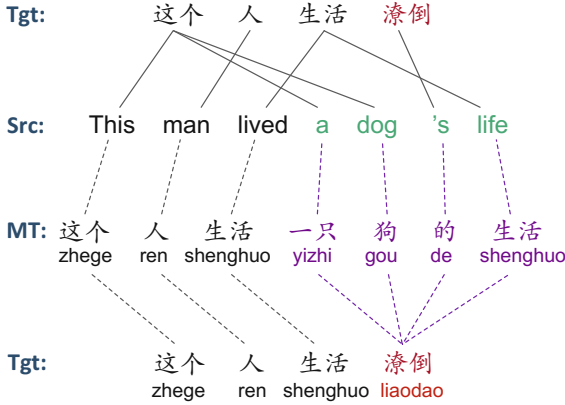


Fig. 1. An example of wrong alignment leading to bad rule-extraction. The alignment between the first and second line is the result of Giza++ [14]. The second alignment between “Src” and “MT” is done by MT decoding. The third alignment is generated by TER-plus [7].

3 Error-Driven Rule Refinement

The process of machine translation could be viewed as a search problem to find the best derivation of translation rules for the input sentence. So the quality of translation rules will greatly affect the performance of the system. On the other hand, the extraction of translation rules is based on word alignment of the bilingual corpus, which is mostly done in an unsupervised manner and the quality is not sufficient [14]. So the alignment error is generally considered as a bottle-neck for rule-extraction [10]. Figure 1 shows an example: the source side *a dog’s life* should be aligned to a specific Chinese term *liaodao*, but since *liaodao* is not a common translation for *dog*, the unsupervised alignment algorithm, i.e. Giza++ [14], will align *dog* to a more common word *zhege* and generate a false alignment between “Src” and “Tgt” in Figure 1. According to the alignment-consistent regulation [9], we will be unable to extract the correct translation rule:

$$a\ dog\ 's\ life \longrightarrow\ liaodao$$

So in the real time translation, we may have to use a more common rule like:

$$a\ dog\ 's\ life \longrightarrow\ yizhi\ gou\ de\ shenghuo$$

This will lead to the wrong translation shown in the third line in Figure 1. These kind of errors are intrinsic for statistical models and hard to avoid. However, with the help of post-editing, we could easily identify those errors and get the right translation. So the motivation of our approach is to learn from the errors discovered in the post-editing process, then generate refined translation rules to correct the errors caused by statistical models. Our method consists of three key parts: error detection, rule extraction and rule filtration, we will give details in the following section.

3.1 Error Detection

Algorithm 1. Error-driven Rule Extraction

```

1: procedure TERP(Hypothesis h, Reference R)
2:    $E \leftarrow \infty$ ,  $Ops \leftarrow \{\}$ 
3:   for  $r \in R$  do
4:      $h' \leftarrow h$ ,  $e \leftarrow 0$ 
5:     repeat
6:       Find operation s, that most reduces min-edit-distance(h, r)
7:       if s reduces edit distance then
8:          $h' \leftarrow \text{apply } s \text{ to } h'$ 
9:          $e \leftarrow e + 1$ ,  $p \leftarrow s$ 
10:      end if
11:     until No operation that reduces edit distance remain
12:      $e \leftarrow e + \text{min-edit-distance}(h, r)$ 
13:     if  $e < E$  then
14:        $E \leftarrow e$ 
15:        $Ops \leftarrow p$ 
16:     end if
17:   end for
18:   return  $E, Ops$ 
19: end procedure
20:
21: procedure RULE EXTRACTION(Operations ops)
22:   rule-set  $\leftarrow \{\}$ 
23:   for  $p \in ops$  do
24:      $t, t' \leftarrow p$ 
25:      $s \leftarrow \text{FindSource}(t)$ 
26:     rule-set  $\leftarrow \langle s, t' \rangle$ 
27:   end for
28:   return rule-set
29: end procedure

```

We use the SiPE framework described in Section 2 to simulate the post-editing process and generate translation-reference pairs as “MT” and “Tgt” in Figure 1. To measure the absolute difference between the two strings, we use Translation Error Rate Plus [7], which is an edit-distance based metric and an extension of TER [21] tailed for machine translation evaluation. Since the correct translations may differ not only in lexical choice but also in the order in which the words occur, TERp allows block movement of words, called shifts, within the hypothesis. Shifting a phrase is assumed to have the same edit cost as inserting, deleting or substituting a word, regardless of the number of words being shifted. This metric correlates well with the translation quality and could provide not only the score but also the edit operations needed to exactly transform the output into the reference. The pseudo-code is shown in the TERp procedure of Algorithm 1.

3.2 Rule Extraction

One advantage of TERp is that it generates an adequacy score by penalizing deletions, insertions, substitutions and shifts. This often allows it to calculate a set of shifts that largely align MT output to a reference, even when MT output uses significantly long orderings. This trait helps us to get larger chunk of modification rules rather than massive small pieces of rules which is hard to use in real time decoding. As shown in the last two lines in Figure 1, we apply TERp procedure on SMT output (“MT”) and the reference (“Tgt”), and only one edit operation is needed:

$$yizhi\ gou\ de\ shenghuo \longrightarrow liaodao$$

Given the operation, we could perform rule extraction. The procedure straightforward: during decoding, we could align each source phrase s with its translation t , then using TERp procedure we could obtain the right modification t' of t . So it's easy to align the source side s (*a dog's life* in Figure 1) with the correct translation t' (*liaodao*), generating the correct translation rule. Shown in Rule-extraction procedure in Algorithm 1.

However, the pending problem is that the translation probability of the new rule is hard to estimate. Since it would be very expensive and time consuming to modify the alignment and re-calculate the probability over the whole training-set, we propose two schemes for probability estimation:

First we could heuristically set a high probability, assuming that all the newly learned rules should be preferred in translation.

Due to the complexity of MT errors, the generated rules may not be of high quality, further more, manually-set score may break the overall balance of the model, resulting in new errors in other translation scenarios. So we introduce a more balanced scheme by treating both original rule and the new rule equally, which allows the other features in SMT such as language model to determine which rule to use in real-time decoding.

The experimental results show that the second scheme achieves better performance, and the first scheme in certain cases will hurt the system.

3.3 Context-Based Rule Filtration

Since the quality of SMT output is relatively poor, a large number of modification rules will be generated based on our method. But due to the complexity of translation errors, some bad rules could also be generated. To address this issue, we propose a simple but effective rule-filtering method which use rule context to determine the goodness of the modification rule. We define the context of the rule C by the number of identical surrounding words, and P denotes the number of words within the rule. So C ensures the stable context of rule and filter out rules with unfaithful translations, And P will filter out too long modification rules which are unlikely to be used in test-set. In our experiment, we heuristically adjust C and P to obtain the best quality modification rules. And the best performance is achieved by setting $C \geq 2$ and $P \leq 5$.

4 Experiment

4.1 System Preparation and Data

To testify the solidness of our method, We conduct Chinese-to-English translation experiments on two different domains: news domain and medical domain, the information of the corpus is shown in Table 1. For comparison, We introduce two baselines:

1. Moses: a state-of-art phrase-based SMT system [15], available online¹. we use the standard 11 features, set beam size to 200, max-phrase-length to 7, and distortion limit to 6
2. Hiero: an in-house implementation of Hierarchical Phrase-Based (HPB) model [16]. we use basic 8 features, and set beam size to 300, max-phrase-length to 7.

We word-aligned the training data using GIZA++ with refinement option “grow-diag-and” [17], and trained 4-gram language model on giga-xinhua corpus using the SRILM toolkit [20] with modified Kneser-Ney smoothing. For parameter tuning, we use minimum error-rate training [12] to maximize the Bleu score on the development set. We evaluate translation quality using case-insensitive Bleu-4, calculated by the script `mteval-v11b.pl`. We also report the TERp scores calculated by TER-Plus [7].

Table 1. Overview of the data-sets used in the experiment. All the numbers are sentence count.

Domain	Training-set	Dev-set	Test-set		
News	240k	Nist02	Nist04	Nist05	Nist06
Chemistry	560k	1000	1000		

4.2 Results and Analysis

We first show the results on news domain in table 2: “heuristic” and “balanced” denotes the two schemes for assigning translation probability to refined rules. Since “heuristic” assigned high probability to refined rules, they were always preferred in decoding, thus hurting the system by breaking the balance of the statistical model. On the other hand, “balanced” scheme assigned the same probability with the original rule, so in real time decoding, other SMT features like language model could determine the right rule to use. For phrase-based system, our method gains an average improvement of 1.42 bleu points over all test-sets. But for hierarchical phrase-based system, the improvement is relatively small. The reason is two-folded: first HPB model generates hierarchical rules which could

¹ www.statmt.org/moses/

Table 2. Final results on news domain, the number in bold means the improvement is statistically significant ($p < 0.05$)

System	Bleu				TERp			
	04	05	06	avg	04	05	06	avg
moses	32.02%	29.00%	27.18%	29.34%	61.47%	64.04%	66.45%	64.47%
heuristic	31.73%	28.66%	26.22%	28.87%	62.23%	64.67%	67.03%	64.64%
balanced	33.47%	30.02%	28.80%	30.76%	60.24%	59.37%	63.89%	61.77%
hiero	34.10%	29.89%	28.78%	30.92%	59.55%	62.73%	64.84%	62.37%
balanced	34.09%	29.87%	28.81%	30.92%	59.56%	62.75%	64.85%	62.38%

greatly expand the rule coverage, which compensate the effect of bad alignment. The second reason may be that the alignment quality of the news domain is relatively good with fewer rare word so the rule-extraction errors is not very severe.

We also show the TERp score in the last column, and it’s reasonable that the performance is in accordance with bleu. The average drop is TERp score is about 2.7.

The results on medical domain is also promising, shown in Table 3. We can see that on phrase-based model the bleu gains is 0.51 point, at the same time, for hirarchical phrase-based model the improvement is 0.78, much more significant than that on the news domain. This is because there are many formula and special terms in medical-domain corpus which makes it hard for unsupervised alignment, causing more errors in rule extraction. So our method produced more significant improvement by generating better translation rules.

Table 3. Performance on testset of medical domain

System	BLEU	TERp
moses	29.64%	66.06%
Ours	30.15%	63.80%
hiero	29.48%	63.53%
Ours	30.26%	62.57%

The effect of rule filtration is also critical to our approach. Since there are still some noise in the generated rules, we tried different heuristic filter settings to test the performance of the system. Figure 2 shows the results, we can see that more strict filtration settings produced better performance: adding 0.5 million new rules degrades the performance a little bit, then we gradually constrain the filtration settings and the performance gets better with the peak of 0.7 bleu point gain.

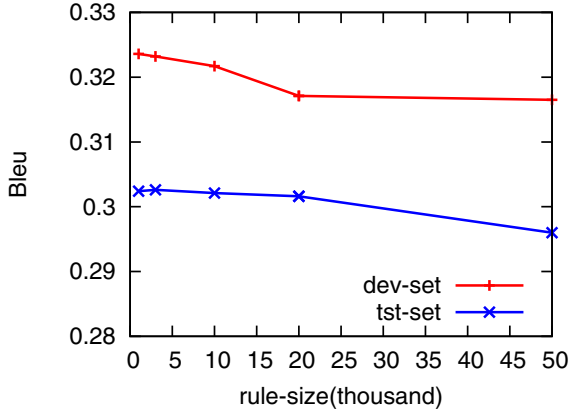


Fig. 2. The impact of different filtration heuristics on Bleu score

5 Related Work

Our work belongs to the family of Statistical Post-editing [1], which intends to use statistical method to capture the errors made by the translation system to further improvement performance. Simard et.al (2007) and Lagarda et.al (2009) both use a phrase-based SMT system to post-edit the output of a rule-based system, this method could combine the merits of both models and yield significant improvement. Bechara et.al (2011) is the first to directly use a SMT system to post-edit another SMT system, the key part of their success is that their post-editing system uses additional corpus to train. Our method is very different from theirs in that rather than relying on another powerful system, we try to dynamically improve the original system without bringing in more complex models and additional error. Besides, our method don't require any additional resources and learn directly from SMT training data.

There are also some work focus on utilize post-editing techniques to improve MT. Isabelle et.al (2007) use automatic post-editing to solve domain adaptation problem in MT. Mundt et.al (2012) learn to automatically recover dropped content words from post-editing. And Denkowski et.al (2014) use post-editing to train an online adaptation framework for SMT. Our work is in the same spirit with theirs, but we focus on rule refinement task.

The simulated post-editing paradigm in our work could also be viewed as a force decoding process [23, 24], in which we can boost new translation rules for better forced decoding. The difference is that we don't require a strict forced decoding, which is too strict for some MT cases, but try to detect errors in the process and generate refined rules.

6 Conclusion

In this paper we have introduced a novel rule refinement method for SMT. We use a simulated post-editing paradigm to efficiently collect the training data. And use TER-Plus for translation error detection and modification rule-extraction. Finally, to ensure the goodness of the generated rules, we introduce a simple and effectively heuristic algorithm for rule-filtration. We apply our method to both phrase-based and syntax-based SMT systems and gains an overall improvement of 1.4 BLEU point without using any additional resources. In the future, we will try to test our method on more complex translation models and produce more powerful feedbacks to improve SMT systems.

Acknowledgement. We thank the three anonymous reviewers for helpful suggestions. The authors were supported by CAS Action Plan for the Development of Western China (No. KGZD-EW-501) and National Natural Science Foundation of China (No. 2012BAH39B03). Qun Liu's work was partially supported by the Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the CNGL at Dublin City University. The views and findings in this paper are those of the authors and are not endorsed by the Chinese governments.

References

1. Simard, M., Goutte, C., Isabelle, P.: Statistical phrase-based post-editing. In: Proceedings of NAACL (2007)
2. Bechara, H., Ma, Y., van Genabith, J.: Statistical post-editing for a statistical MT system. In: Proceedings of MT Summit XIII, pp. 308–315 (2011)
3. Lagarda, A.L., Alabau, V., Casacuberta, F., et al.: Statistical post-editing of a rule-based machine translation system. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion, vol. Short Papers, pp. 217–220. Association for Computational Linguistics (2009)
4. Dugast, L., Senellart, J., Koehn, P.: Statistical post-editing on SYSTRAN's rule-based translation system. In: Proceedings of the Second Workshop on Statistical Machine Translation, pp. 220–223. Association for Computational Linguistics (2007)
5. Denkowski, M., Dyer, C., Lavie, A.: Learning from Post-Editing: Online Model Adaptation for Statistical Machine Translation. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (2014)
6. Navarro, G.: A guided tour to approximate string matching. *Journal of ACM computing surveys (CSUR)* 33(1), 31–88 (2001)
7. Snover, M.G., Madnani, N., Dorr, B., et al.: TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. *Journal of Machine Translation* 23(2-3), 117–127 (2009)
8. Hardt, D., Elming, J.: Incremental Re-training for Post-editing SMT. In: Proceedings of AMTA (2010)

9. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Journal of Computational linguistics* 29(1), 19–51 (2003)
10. Liu, Y., Xia, T., Xiao, X., et al.: Weighted alignment matrices for statistical machine translation. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 2, pp. 1017–1026. Association for Computational Linguistics (2009)
11. Brown, P.F., Cocke, J., Pietra, S.A.D., et al.: A statistical approach to machine translation. *Journal of Computational linguistics* 16(2), 79–85 (1990)
12. Och, F.J.: Minimum error rate training in statistical machine translation. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, vol. 1, pp. 160–167. Association for Computational Linguistics (2003)
13. Papineni, K., Roukos, S., Ward, T., et al.: BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics (2002)
14. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. *Journal of Computational linguistics* 30(4), 417–449 (2004)
15. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*, demonstration session, Prague, Czech Republic (2007)
16. Chiang, D.: Hierarchical Phrase-Based Translation. *Journal of Computational Linguistics* 33(2), 201–228 (2007)
17. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL 2003)*, vol. 1, pp. 48–54. Association for Computational Linguistics, Stroudsburg (2003)
18. Mundt, J., Parton, K., McKeown, K.: Learning to Automatically Post-Edit Dropped Words in MT. In: *Proceedings of AMTA* (2012)
19. Isabelle, P., Goutte, C., Simard, M.: Domain adaptation of MT systems through automatic post-editing. In: *Proceedings of MTS* (2007)
20. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: *Proceedings of Intl. Conf. on Spoken Language Processing*, Denver, vol. 2, pp. 901–904 (2007)
21. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: *Proceedings of Association for Machine Translation in the Americas*, pp. 223–231 (2006)
22. Niessen, S., Och, F., Leusch, G., Ney, H.: An evaluation tool for machine translation: fast evaluation for MT research. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pp. 39–45 (2000)
23. Yu, H., Huang, L., Mi, H., Zhao, K.: Max-Violation Perceptron and Forced Decoding for Scalable MT Training. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1112–1123 (2013)
24. Liang, H., Zhang, M., Zhao, T.: Forced decoding for minimum error rate training in statistical machine translation. *Journal of Computational Information Systems* (8), 861868 (2012)

Case Frame Constraints for Hierarchical Phrase-Based Translation: Japanese-Chinese as an Example

Jiangming Liu¹, JinAn Xu^{1,*}, Jun Xie², and Yujie Zhang¹

¹ School of Computer and Information Technology, Beijing Jiaotong University

² Institute of Computing Technology Chinese Academy of Science
{jaxu,yjzhang}@bjtu.edu.cn, {jmliuunlp,stiffxj}@gmail.com

Abstract. Hierarchical phrase-based model has two main problems. Firstly, without any semantic guidance, large numbers of redundant rules are extracted. Secondly, it cannot efficiently capture long reordering. This paper proposes a novel approach to exploiting case frame in hierarchical phrase-based model in both rule extraction and decoding. Case frame is developed by case grammar theory, and it captures sentence structure and assigns components with different case information. Our case frame constraints system holds the properties of long distance reordering and phrase in case chunk-based dependency tree. At the same time, the number of HPB rules decrease with the case frame constraints. The results of experiments carried out on Japanese-Chinese test sets shows that our approach yields improvements over the HPB model (+1.48 BLEU on average).

1 Introduction

The hierarchical phrase-based (HPB) model (Chiang, 2007) is widely used in statistical machine translation. Extended from phrase-based (PB) rules (Koehn et al., 2003), HPB rules are capable of capturing phrase-level reordering by exploiting the underlying hierarchical structures in natural language. HPB model is formally synchronous context-free grammar but this is learned from a bitext without any syntactic information, so that HPB suffers from limited phrase reordering in the case of combining translated phrases with monotonic glue rules. As a result, it performs not so well in long distance reordering. Furthermore, without phrase boundary determination, the number of HPB rules increases explosively with the increase in training data. To address the HPB model limitation, a number of work is motivated in two aspects.

In the process of HPB decoding, many recent work are motivated to preserve linguistic information in HPB model derivation. Syntactic features are derived from the source dependency parsing to directly guide derivation in HPB model (Marton and Resnik, 2008; Huang et al., 2010; Gao et al., 2011; Marton et al., 2012). However, these systems perform not so well in agglutinative language

* Corresponding author.

translation due to the agglutinative properties of complex and varied morphology. The agglutinative languages are usually provided with relatively accurate chunk-based dependency analysis. The structure impedes the utilization of word-based dependency to string translation model (Flannery, et al. 2011).

In terms of extraction and presentation of HPB rules, many significant works focus on assigning HPB rules with extra constraints to explore search space (Li et al., 2012; He et al., 2010), and the suited HPB rules can be selected from the rule selection model (Liu et al., 2008; He et al., 2008). However, the total number of HPB rules remains the same, and a large number of redundant rules are extracted.

To solve the above two problems, we exploit case frame constraints (CFCs) in this paper. The description of case frame will be introduced in section 3. At the same time, this paper presents case chunk-based dependency, and the purposes of our works include alleviating reordering problem and restricting HPB rule extraction in case frame, and finally case frame HPB rules (CF-HPBs) are extracted. In terms of the reordering process, case frame reordering rules (CF-Rs) are automatically extracted from the source side parsing and aligning parallel corpus, and this aims to alleviate the reordering problems under the condition of preserving all the components in the sentence.

This paper proposes a novel approach to use case frame constraints in Japanese-Chinese statistical machine translation as an example and achieve better performance than HPB model and word-based dependency model as shown in our experimental results.

According to our knowledge, case frame is rarely used in statistical machine translation. Our work is the first to try case frame in statistic machine translation. The main contributions of our works are using case frame constraints in HPB rule extraction and decoding.

The remainder of this paper is organized as follows. Section 2 introduces some related work and mainly contributes to this paper. We present case frame constraints HPB rules (CF-HPBs) extraction in section 3, then we define the case chunk-based dependency tree and describe case frame reordering rules in section 4. Section 5 presents our model. Section 6 reports our experiments. Section 7 presents the analysis on the experimental. Section 8 concludes this paper with prospects for future work.

2 Related Work

In recent years, word-based dependency structure is widely used to incorporate linguistic information into machine translation (Lin, 2004; Quirk et al., 2005; Ding and Palmer, 2005; Xiong et al., 2007). The reordering problem can be alleviated, especially in long distance reordering problem (Xie et al., 2011). Dependency-to-string model employs rules whose source-side is a word-based dependency structure with POS and target as string. Reordering problem can be alleviated by simple nodes exchange.

Many novel approaches are presented for restricting HPB rules extraction (He et al., 2010; Xiong et al., 2010). These methods employ supervised learning

technology, and suitable features are selected to train a boundary classifier as soft constraint for decoding. However, a total number of HPB rules is not decreased and a large-scale corpus is needed for training classifier.

Our proposed approach focuses on case frame constraints (CFCs) to improve the quality of extracted rule and decoding. Moreover, this paper defines a new structure dependency tree, which is more suitable for agglutinative language than word-based dependency tree. The derivation based on new structure tree holds two merits. Firstly, the number of HPB rules is decreased. Secondly, the decoding efficiency and translation quality are improved.

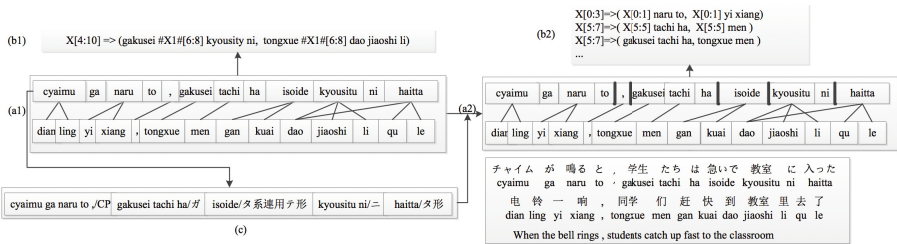


Fig. 1. An example of phrase boundary determination for CF-HPBs extraction

where (a1) is sentence pair with word alignments; (b1) is an example of HPB rules without case frame constraints; (c) is source side sentence with case boundary generated by KNP tools; (a2) is a sentence pair with word alignment and case boundary (marked in bold); (b2) is a set of CF-HPBs examples.

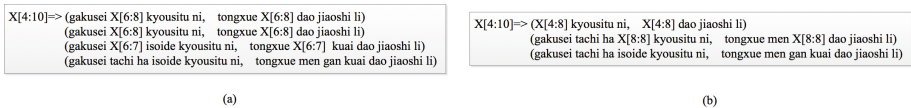


Fig. 2. Examples of unreasonable derivation (a) and reasonable derivation (b)

3 Case Frame Constraints

3.1 Case Frame

Case grammar created by Fillmore (1968) and developed by Cook (1989) in English case grammar, is used to linguistically analyze the surface syntactic structure of sentences by investigating the combination of cases. Case frame is analyze by case grammar. Case grammar has been developed in different languages. In Japanese, a case frame corpus is extended and built from web resources (Kawahara and Kurohashi, 2006). Under the case frame corpus, the system of Japanese syntactic and case structure analysis turns to be a state-of-the-art (Buchholz and Marsi, 2006).

3.2 CFCs on HPB Rules

HPB rules replaces common phrase with non-terminal variables, which confuses primary with secondary linguistic. A large number of long HPB rules are slightly linguistic, especially in the case of the Japanese language.

Let $S = (sw_0, sw_1, \dots, sw_l)$ be a source word sequence and $T = (tw_0, tw_1, \dots, tw_m)$ be a target sequence, where sw_i and tw_j are source word and target word respectively. With word alignment, HPB rules can be extracted as:

$$\begin{aligned} &(sw_0 X_{(1) \rightarrow (0)} sw_2, X_{(1) \rightarrow (0)} tw_1), \\ &(sw_0 X_{(1..2) \rightarrow (0..1)} sw_3 sw_4, X_{(1..2) \rightarrow (0..1)} tw_2 tw_3), \\ &(sw_0 X_{(1..3) \rightarrow (0..2)} sw_4, X_{(1..3) \rightarrow (0..2)} tw_3), \end{aligned}$$

where X is non-terminal variable and its indices denotes the word alignment. The non-terminal variables in HPB rules can be generated by replacing common phrase without distinguishing what the component means. To make derivation more reasonable, we use syntax to assign components with specific semantic information that makes sense.

According to case frame theory, a sentence is divided into many components. CFCs are used in determining component boundaries during HPB rule extraction, at the same time, each component can be labeled with specific case information. Each case boundary is regarded as the phrase boundary in the process of HPB rule extraction. Suppose a case frame in source language given like $CF = \{(sw_0)_{subject}, (sw_1 sw_2)_{verb-head} (sw_3 sw_4)_{object} \dots\}$ where the phrase sw_0 is marked as subjective case, phrase $sw_{1..2}$ as head, and $sw_{3..4}$ as objective case. The traditional HPB rule with the non-terminal variable $X_{(1..3) \rightarrow (0..2)}$ is filtered due to the fact that $X_{(1..3)}$ is over verb-head case boundary (phrase boundary). As a result, phrases without over the case boundary (phrase boundary) can be generalized as non-terminal variable. It means that many rules without suitable to case frame are filtered, and finally CF-HPBs will be achieved semantically. An example is shown in Figure 1, where “tongxue men” means students, and “gan kuai dao” means “catch up fast”. “men gan kuai” is unreasonably generalized for a non-terminal variable in Figure 1(b1). Also, in case frame constraints, “men gan kuai” is aligned to source side sequence “tachi ha isoide” over the case boundary that is forbidden in CF-HPBs, and then it will be filtered. In this way, each component in CF-HPBs can be assigned with a semantic label, namely case.

The extra properties of CF-HPBs are maintained bellow:

Property 1. *An acceptable non-terminal variable is only generalized by the phrase without over the component boundary (case boundary).*

Property 2. *An acceptable non-terminal variable can be generalized by any common phrase inside one component.*

Due to these properties, a reasonable derivation can be obtained as shown in Figure 2, which is part of derivation in the case of Figure 1.

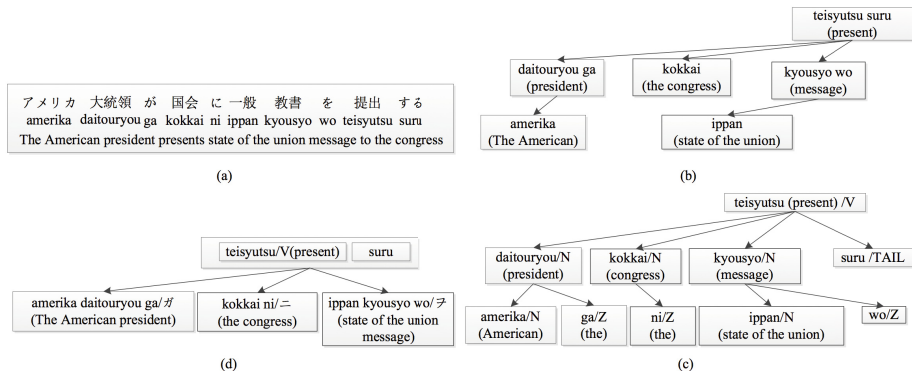


Fig. 3. Example of chunk-based dependency structure and word-based dependency structure

where (a) is the original source side sentence; (b) is a chunk-based dependency structure given by KNP; (c) is a word-based dependency structure with a simple change from chunk-based structure and (d) is a case chunk-based dependency structure, where “The American president” is subjective denoted by specific case tag, “the congress” is objective and “state of the union message” is direction

4 Case Frame Reordering

4.1 Case Chunk-Based Dependency

Before extraction of case frame reordering information, a specific structure, namely case chunk-based dependency structure, is firstly defined.

In case frame, a sentence can be divided into many components with different cases, where each component is a word or a phrase, which is defined as a chunk in this paper. Case chunk-based dependency tree can be defined as a tuple $CT = (\Sigma, A, C, D)$, where Σ is a set of words, A is a set of chunks corresponding to case boundary, C is a set of possible case tag for chunks, and D is dependency relation² between chunks. It is distinguishable for case chunk-based dependency tree with word-based dependency tree shown in Figure 3. In case chunk-based dependency tree, each node consists of chunk and related case tag.

Due to parallel sentence pairs given for statistical machine translation, in our model, source side sentence is represented by case chunk-based dependency structure and target side sentence is represented by word sequence. With word alignments, our initial model is defined by a tuple (CT, Δ, A) where CT is source side case chunk-based dependency tree, Δ is a set of target side words and A is word-to-word alignment. An example is shown in Figure 4(a). Since source side item (node) is chunk-level and target side item is word-level, the change is carried out from word-to-word alignment A to chunk-to-word alignment A' .

$$A' = \{(c, tw) | \exists sw \in c, (sw, tw) \in A, c \in \Lambda, tw \in \Delta\}$$

So our final model is defined by $M = (CT, \Delta, A')$. Based on the model, case frame reordering rules can be extracted.

4.2 Case Frame Reordering Rule

Case frame reordering rules (CF-Rs) are represented by a tuple (t, s, \sim) where:

- t is a dependent relation of the source dependency structure, with each node labeled with a variable from a set $X = \{x_1, x_2, \dots\}$ constrained by a case from C . Specially, head node can be also labeled by a chunk constrained by a case from C .
- $s \in X$ are the target side chunk slots corresponding to source side non-terminal variables.
- \sim is a one-to-one mapping from slot in s to variables in t

One example is shown in Figure 4(c).

According to our rule definition, CF-Rs have two properties bellow:

Property 3. *Each node, except head node in source side of rules, is unlexicalized, and each item in target side is slot with variable corresponding to source side variable.*

Property 4. *Head node in source side can be lexicalized or unlexicalized. And thus CF-Rs can be classified into CF-LRs (lexicalized) and CF-URs (unlexicalized).*

Prior work on rule extraction, reordering and lexical translation are both considered at the same time. Also, alignment error propagation impacts reordering rule and lexical translation. Instead, during the extraction process of CF-Rs, we only consider the variables reordering on the target side. In the following section, we will present how to extract rules using our model.

4.3 CF-Rs Acquisition

Now, it focuses on reordering rule extraction. Before extraction, anchor is defined to assist reordering formation extraction among each item on the target side. Anchor can be represented as a function $Ach(sp)$, where $sp \in SP$ denotes possible span on the target side. Here, span is a set of word index on the target side corresponding to certain node on the source side tree (same index may occur twice or more), where spans of all child nodes are at chunk-level and spans of head nodes are at word-level. So the amount of spans is larger than the number of nodes on the source side tree. In Figure 4(b), the head node has two spans and each child node has only one span. The value of $Ach(sp)$ is a real number which is computed using following formula

$$Ach(sp_i) = sum(Sign(sp_i, SP_i))$$

Where sp_i denotes the i th span in SP , SP_i denotes set of spans except sp_i , $Sign(sp_i, SP_i)$ returns a $|SP_i|$ size vector (V_i) where each item is 1 or 0 and

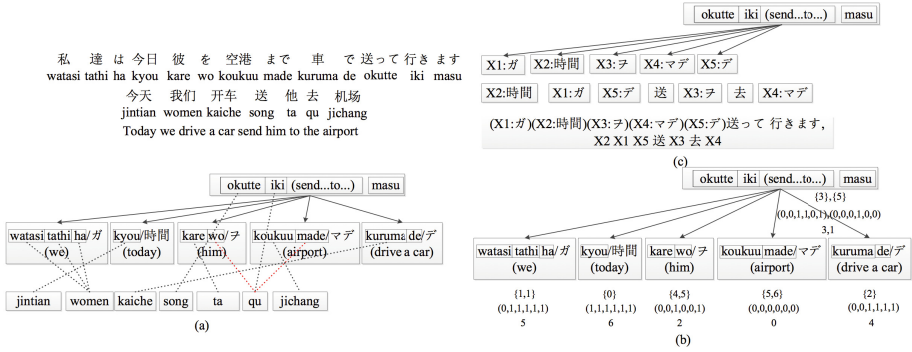


Fig. 4. An example of CF-Rs extraction

where (a) is a case chunk-based dependency-to-string with word alignment, where the red dotted line is noise alignment; (b) is with chunk alignment, where each node has three extra items in extraction, first is spans, second is signal vector, and third is *anchor*; (c) is an extracted rule.

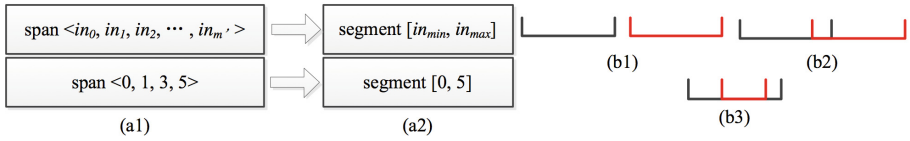


Fig. 5. Span, segment and three spans relation

where (a1) is a span. (a2) is a segment from the span. (b) is a separation span relation, (c) is an intersection span relation and (d) is an inclusion span relation, where each of them is a segment.

sum is a function to sum up all item in that vector. $V_i = (v_0, v_1, \dots, v_j, \dots, v_{|S P_i|})$ where according to chunk-word alignment, v_j is 1 if and only if the j^{th} span is relatively left to the i^{th} span, otherwise it is 0. To formulate the span relation, function $F(i, j)$ is defined to capture relation between i^{th} spans and j^{th} span. $F(i, j)$ follows three strategies, which respectively deal with three situations.

- **Separation** segment i is separated from segment j , where segment i is generated by minimum index and maximum index in sp_i , and segment j is similar. Under this condition as shown in Figure 5(b), $F(i, j)$ is 1 if and only if largest index in sp_i is smaller than or equal to smallest index in sp_j , otherwise it is 0.
- **Intersection** segment i and segment j are interacted. Under this condition as shown in Figure 5(c), $F(i, j)$ is 1 if and only if sum of all index in sp_i is smaller than in sp_j , otherwise 0.
- **Inclusion** segment i cover segment j , or segment j covers segment i . Under this condition as shown in Figure 5(d), $F(i, j)$ is 0, which means the default value remains the same.

Generally speaking, each node on the source side will place its own non-terminal variable on the target side with left-right order according to its anchor (may also be called a rank). Briefly speaking, Anchor ensures the order of target side items corresponds to the source side items. In this way, CF-Rs are extracted as shown in Figure 4(c).

CF-Rs are similar with dependency-string rules as mentioned in (Xie, et al. 2011). However, CF-Rs are guided by a case frame, and their semantic labels considers case frame structure in the whole sentence, conversely, POS only consider one word and components are neglected in the whole sentence. Moreover, some alignment errors will be alleviated in obvious anchor function. Under case constraints, fuzzy reordering information extraction is useful in agglutinative language due to its complex morphemes.

Following the case grammar, each component in a sentence will have a complete semantic representation. CF-Rs only achieve reordering information among components. Translation inside components is done using CF-HPBs as described in previous section.

Generally, in case frame, outside reordering of each component in sentence is done using case frame. And then, inside translation each component it is done using phrase-based rules, which is superior in HPB model in terms of short-distance reordering and lexical translation.

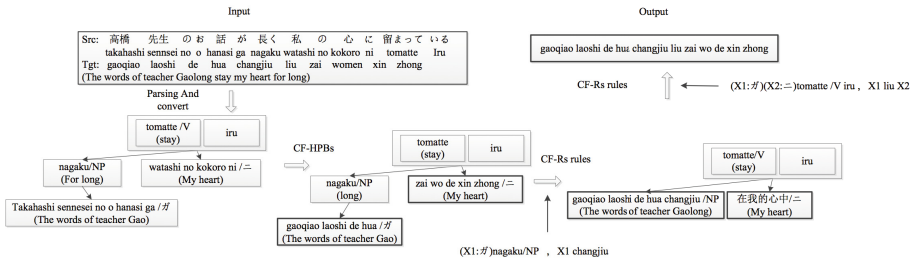


Fig. 6. An example of derivation

where in each step, the bolded box represents current translation focus.

5 Translation

5.1 Derivation

In this paper, CF-HPBs and CF-Rs are defined, and are integrated in derivation, where both CF-HPBs and CF-Rs are named case frame rules. This subsection describes one possible derivation in details as shown in Figure 6.

Under the case frame, a sentence is decomposed into a number of components, each of which has completed semantic content. Furthermore, a sentence is expressed in a form of a case chunk-based dependency tree. For one derivation, CF-HPBs are used in inside components derivation, and CF-Rs are used in outside components derivation, i.e. components reordering.

5.2 Log-Linear Model

Following (Och and Ney, 2002), we adopt a general log-linear model. Let d be a derivation that translate source chunk-based dependency tree T into a target string e . The probability of d is defined as:

$$P(d) \propto \prod_i \phi_i(d)^{\lambda_i}$$

Where ϕ_i are features defined on derivations and λ_i are feature weights. Due to the fact that our translation rules have two sets, namely CF-HPBs and CF-Rs, during derivation, two kinds of rules are integrated. In our experiments of this paper, we used nine features which are similar with (Xie et al., 2011) as follow:

- CR-HPBs translation probabilities $P_{HPB}(t|s)$ and $P_{HPB}(s|t)$;
- CR-HPBs lexical translation probabilities $P_{(HPB)lex}(t|s)$ and $P_{(HPB)lex}(s|t)$;
- CF-Rs translation probabilities $P_R(t|s)$ and $P_R(s|t)$;
- CF-Rs lexical translation probabilities $P_{Rlex}(t|s)$ and $P_{Rlex}(s|t)$;
- Rule penalty $exp(-1)$;
- Language model $P_{lm}(e)$;
- Word penalty $exp(|e|)$.

During feature tuning process, different features are added into log-linear model and each weight of features can be discriminatively trained by MERT (Och, 2003), which is similar to (Li et al., 2012; Xie et al., 2011). Features include translation probabilities, lexical translation probabilities, language model, rule penalty, and word penalty.

5.3 Decoding

Our decoder is based on bottom up chart parsing. It determines the best derivation d^* that translates the input case chunk-based dependency structure into a target string among all possible derivations D : $d^* = argmax_{d \in D} P(D)$

Given a source case chunk-based dependency structure T . For each accessed internal node n , it gets a case frame corresponding to the node n , and checks if the CF-Rs are set for the matched reordering rules, and then checks if the CF-HPBs rule is set for matched translation rules. If there is no matched rule, we construct a *pseudo translation rule* according to the case frame, which has no reordering information like glue rules. Due to a large search space, a large number of translation is generated by substituting the variables in the target side of a translation rule with the translations of the corresponding slots in the source case frame. Similar to (Xie, et al. 2011), we make use of cube pruning (Chiang, 2007; Huang and Chiang, 2007) to find candidates with integrated language model for each node.

6 Experiments

We evaluate the case frame constraints in the replications of hierarchical phrase-based model in Japanese-Chinese translation. In these experiments, a replication

of hierarchical phrase-based model is taken as a baseline model with beam size is 200 and the beam threshold of 0. The maximum initial phrase length is 10. In order to compare chunk-based dependency and word-based dependency, we also take dependency to string (*dep2str*) system by simply changing from chunk dependency to word dependency in word-POS process as shown in Figure 3. Under the same condition, this paper utilizes our model to constrain rule extraction and decoding.

6.1 Data

Due to that Japanese-Chinese parallel corpus is rare, our corpus consists of 280k sentence pairs for training which come from CWMT 2011 (Zhao et al., 2011) Japanese-Chinese evaluation task data in news domain. 500 sentence pairs are for parameters optimization. For testing, we use 900 sentence pairs provided by the task. In addition, we mix all the sentence pairs (including training, developing and testing data), and randomly select 500 sentence pairs for developing, 900 sentence pairs for testing and the rest of the sentences for training.

The source side sentences are parsed by KNP (Kurohashi and Nagao, 1994) into chunk dependency structures whose nodes are at chunk-level. Also we achieve corresponding case frame analysis from byproduct of KNP. The word alignment is obtained by running GIZA++ (Och and Ney, 2003) on the corpus in both direction and applying “grow-diag-and” refinement (Koehn et al., 2003). We apply SRI Language Modeling Toolkit (Stolcke, 2002) to train a 5-gram language model for target side sentences.

6.2 Baseline Model

In order to evaluate our system performance, we take a replication of Hiero (Chiang, 2007) as the hierarchical phrase-based model baseline (*hiero-re* for short), where we set the beam size $b = 200$ and the beam threshold $\beta = 0$. The maximum initial phrase length is 10.

Also, we use *dep2str* as the dependency-to-string model baseline, which consider word based dependency as provided by (Xie et al., 2011), where the same parameters are used for the experiment.

6.3 Result

Table 1 illustrates the translation experimental results. It shows that our system has achieved the best results on test sets, with +2.83 BLEU points on average higher than that of *dep2str*, and +1.23 BLEU points on average higher than that of *hiero-re*. It demonstrates that case frame constraints are useful to improving translation quality for HPB model. Compared with *dep2str*, chunk-based dependency tree performs better than word-based dependency does. In terms of the rule amount, the number of CF-Rs and CF-HPBs is decreased by more than half in the corpus of 280k sentence pairs. We believe case frame constraints superiority can be more obvious in larger corpus.

Table 1. The BLEU-4 score (%) on test sets of different system

System	Rule#	CWMT Mix		Avg
<i>hiero-re</i>	24.0M	22.26	18.46	20.33
<i>dep2str</i>	2.8M	19.34	18.12	18.73
ours	1.4M+10.0M	22.62*	20.50*	21.56*

where the “+” denotes the 1.4 million CF-Rs and 10 million CF-HPBs on case frame constraints. The “*” denotes that the results show significant improvements over all of the other systems (p<0.01)

7 Analysis

In HPB model, glue rule is frequently used for combining long sub-sentence without considering possible reordering. The agglutinative language, Japanese for example, has complex and varied morphology. Although the utilization of POS is general for the dependency rule variables in *dep2str*, it has local lexicalization, and some translation words are omitted. CF-HPBs maintain phrase translation with semantic label and CF-Rs alleviate long distance reordering problem. To further our analysis, we compare some actual translations generated by *hiero-re*, *dep2str* and our system. Figure 7 give one translation of our test set, which is helpful to elucidate these *problems* in terms of reordering and lexical *translation*.

7.1 Better Reordering

Main structure in Japanese structure is SOV-style, which is different from Chinese SVO-style. Reordering problem is significant in Japanese-Chinese translation, especially with long phrase for S and/or V. Compared with hierarchical phrase-based rules, CF rules have better phrase reordering. In the first example as shown in Figure 7, the source sentence main centered verb chunk is “tuujite (rely on)”, and however, the objective is a long phrase (15 words) depending on the left of that verb chunk, which is a typical SOV-style. *Hiero-re* mistakenly treats that long phrase as subjective, thus results in translation with different meaning from source sentence. Conversely, our system captures this component relations in case frame and translates it into “tuujite (rely on)...”. Although adverb “sarani (further)” is translated with incorrect ordering, the lexical translation is correct, and it makes sense that it cannot influence the understanding of source sentence.

7.2 Better Lexical Translation

Although word-based dependency tree-to-string model can also capture distance reordering problem (Xie, et al., 2011), depending strictly on word alignment in

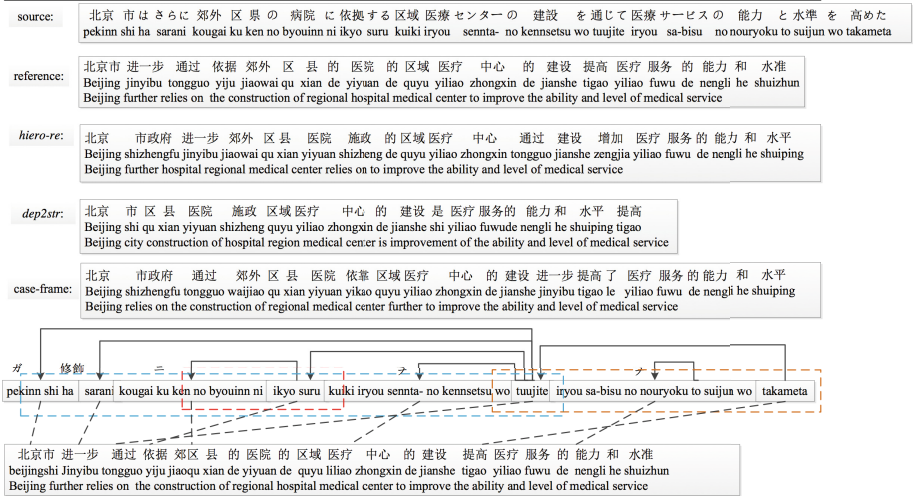


Fig. 7. Actual translations produced by the baselines and our system

For our system, we also display the long distance case chunk-based dependencies correspondence in Japanese and Chinese. In source side dotted box is a case frame.

dep2str, this does not lead to good performance on phrase translation as indicated in Figure 7. The adverb “sarani (further)” and “no (of)” has no corresponding translation or incorrect translation in *dep2str* because they are aligned into a common or NULL. Moreover, complex morphology expressed by long suffixes caused many words to be aligned to incorrect word. Complex alignment brings about some rules that cannot be extracted. Conversely, chunk-based dependency with fuzzy alignment can maintain the phrase-based rule (done with CF-HPBs) extraction of inside components without reordering deficiency (done with CF-Rs).

7.3 Summary

All these results prove the effectiveness of case frame constraints in both long reordering and translation. We believe that case chunk-based dependency tree-string model has an advantage of tending to assign semantic information on variables in rules with case grammar, and not the POS of a word in dependency-to-string model, and also it has an advantage of maintaining phrase structure inside of components with semantic boundary.

The incapability of *hiero-re* in handling long distance reordering is not caused by the limitation of rule representation but by the compromise in rule extraction and decoding for balance between the decoding speed and performance. The hierarchical phrase-based model prohibits any nonterminal X from spanning a substring longer than 10 on the source side that makes the decoding algorithm asymptotically linear-time (Chiang, 2005).

The *dep2str* has a good performance in long distance reordering. However, local lexicalization is restricted by word alignment. Therefore, compatibility with phrases is necessary (Meng, et al. 2013).

8 Conclusion and Future Work

This paper presents case frame constraints for rule extraction and decoding in hierarchical phrase-based model. Compared with HPB rules, the amount of CF-HPBs is decreased. The CF-Rs take the source side as case frame and the target side as string. Our system has an advantage of both long distance reordering and phrase constituency. Moreover, CF-Rs distinguish variables with cases. According to the case frame theory, we interestingly discovered that it can disambiguate some translations. For example, NP with object case or subjective case has different translation.

Case frame constraints are linguistic constraints according to the case grammar theory. It is available for many languages. Case frame can also be used in many aspect of natural language processing, such as summarization, semantic role labeling and bilingual alignment. Meanwhile, more deep semantic case information is expected to further improve the translation quality. Furthermore, It is meaningful to transmit the case information from Japanese to more other languages, and it can be useful to improve the translation quality between more languages.

References

1. Chiang, D.: Hierarchical phrase-based translation. *Computational Linguistics* 33(2) (2007)
2. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: *Proceedings of HLT-NAACL*, pp. 127–133 (2003)
3. Marton, Y., Resnik, P.: Soft Syntactic Constraints for Hierarchical Phrased-Based Translation. In: *Proceedings of the Association for Computational Linguistics*, pp. 1003–1011 (2008)
4. Huang, Z., Cmejrek, M., Zhou, B.: Soft Syntactic Constraints for Hierarchical Phrase-based Translation Using Latent Syntactic Distributions. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 138–147. Association for Computational Linguistics (2010)
5. Gao, Y., Koehn, P., Birch, A.: Soft Dependency Constraints for Reordering in Hierarchical Phrase-Based Translation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 857–868. Association for Computational Linguistics (2011)
6. Marton, Y., Chiang, D., Resnik, P.: Soft syntactic constraints for Arabic-English hierarchical phrase-based translation. *Machine Translation* 26(1-2), 137–157 (2012)
7. Flannery, D., Miyao, Y., Neubig, G., Mori, S.: Training Dependency Parsers from Partially Annotated Corpora. *IJCNLP* 2011, 776–784 (2011)
8. Li, J., Tu, Z., Zhou, G., van Genabith, J.: Head-Driven Hierarchical Phrase-based Translation. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 33–37 (2012)

9. He, Z., Meng, Y., Yu, H.: Maximum entropy based phrase reordering for hierarchical phrase-based translation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 555–563. Association for Computational Linguistics (2010)
10. Liu, Q., He, Z., Liu, Y., Liu, S.: Maximum entropy based rule selection model for syntax-based statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 89–97. Association for Computational Linguistics (2008)
11. He, Z., Liu, Q., Lin, S.: Improving statistical machine translation using lexicalized rule selection. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 2008, pp. 321–328. Association for Computational Linguistics (2008)
12. Lin, D.: A path-based transfer model for machinetranslation. In: Proceedings of Coling 2004, Geneva, Switzerland, August 23–27, pp. 625–630 (2004)
13. Quirk, C., Menezes, A., Cherry, C.: Dependency treelet translation: Syntactically informed phrasal smt. In: Proceedings of ACL 2005, pp. 271–279 (2005)
14. Ding, Y., Palmer, M.: Machine translation using probabilistic synchronous dependency insertion grammars. In: Proceedings of ACL (2005)
15. Xiong, D., Liu, Q., Lin, S.: A dependency treelet string correspondence model for statistical machine translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic, pp. 40–47 (June 2007)
16. Xie, J., Mi, H., Liu, Q.: A Novel Dependency-to-String Model for Statistical Machine Translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 216–226. Association for Computational Linguistics (2011)
17. He, Z., Yao, M., Yu, H.: Learning phrase boundaries for hierarchical phrase-based translation. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 383–390. Association for Computational Linguistics (2010)
18. Xiong, D., Zhang, M., Li, H.: Learning translation boundaries for phrase-based decoding. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 136–144. Association for Computational Linguistics (2010)
19. Fillmore, C.J.: The Case for Case. In: Bach, Harms (eds.) *Universals in Linguistic Theory*, pp. 1–88. Holt, Rinehart, and Winston, New York (1968)
20. Cook, W.A.: *SJ: Case Grammar Theory*. Georgetown University Press, Washington, DC (1989)
21. Kawahara, D., Kurohashi, S.: Case frame compilation from the web using high-performance computing. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, pp. 1344–1347 (2006)
22. Buchholz, S., Marsi, E.: CoNLL-X shared task on Multilingual Dependency Parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 149–164. Association for Computational Linguistics (2006)
23. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of ACL-2003, Sapporo, Japan, pp. 160–167 (July 2003)
24. Zhao, H., Lv, Y., Ben, G., Liu, Q.: Evaluation report for the 7th china workshop on machine translation. In: The 7th China Workshop on Machine Translation, CWMT 2011 (2011)
25. Kurohashi, S., Nagao, M.: Knparsner: Japanese dependency/case structure analyzer. In: Proceedings of the Workshop on Sharable Natural Language Resources, pp. 48–55 (1994)

26. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
27. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada (July 2003)
28. Stolcke, A.: Srilm - an extensible language modeling toolkit. In: *Proceedings of ICSLP*, vol. 30, pp. 901–904 (2002)
29. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. *Proceedings of ACL 2005*, 263–270 (2005)
30. Meng, F., Xie, J., Song, L., Lv, Y., Liu, Q.: Translation with Source Constituency and Dependency Trees. In: *Proceedings of EMNLP 2013*, Seattle, Washington, USA (October 2013)

Bridging the Language Gap: Learning Distributed Semantics for Cross-Lingual Sentiment Classification

Guangyou Zhou¹, Tingting He¹, and Jun Zhao²

¹ School of Computer, Central China Normal University,
152 Luoyu Road, Wuhan 430079, China

² National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
{gyzhou, tthe}@mail.ccnu.edu.cn, jzhao@nlpr.ia.ac.cn

Abstract. Cross-lingual sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of data in a label-scarce target language by exploiting labeled data from a label-rich language. The fundamental challenge of cross-lingual learning stems from a lack of overlap between the feature spaces of the source language data and that of the target language data. To address this challenge, previous work in the literature mainly relies on machine translation engines or bilingual lexicons to directly adapt labeled data from the source language to the target language. However, machine translation may change the sentiment polarity of the original data. In this paper, we propose a new model which uses stacked autoencoders to learn language-independent distributed representations for the source and target languages in an unsupervised fashion. Sentiment classifiers trained on the source language can be adapted to predict sentiment polarity of the target language with the language-independent distributed representations. We conduct extensive experiments on English-Chinese sentiment classification tasks of multiple data sets. Our experimental results demonstrate the efficacy of the proposed cross-lingual approach.

Keywords: Cross-lingual, Sentiment Classification, Deep Learning.

1 Introduction

With the development of web 2.0, more and more user generated sentiment data have been shared on the web. They exist in the form of user reviews on shopping or opinion sites, in posts of blogs or customer feedback in different languages. These labeled user generated sentiment data are considered as the most valuable resources for the sentiment classification task. However, such resources in different languages are very imbalanced. Manually labeling each individual language is a time-consuming and labor-intensive job, which makes cross-lingual sentiment classification essential for this application.

Cross-lingual sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of data in a label-scarce target language by

exploiting labeled data from a label-rich language. The fundamental challenge of cross-lingual learning stems from a lack of overlap between the feature spaces of the source language data and that of the target language data. To address this challenge, previous work in the literature mainly relies on machine translation engines or bilingual lexicons to directly adapt labeled data from the source language to the target language [17,26,24,16,12,27]. Although the machine translation based approaches are intuitive and have advanced the task of cross-lingual sentiment classification, they have certain limitations. First, machine translation may change the sentiment polarity of the original data [9]. For example, the negative English sentence “it is too beautiful to be true” is translated to a positive sentence in Chinese “实在是太漂亮是真实的” by Google Translate (<http://translate.google.com/>), which literally means “it is too beautiful and true”. Second, many sentiment indicative words cannot be learned from the translated labeled data due to the limited coverage of vocabulary in the machine translation results. Recently, Duh et al. [3] report a low overlap between the vocabulary of English documents and the documents translated from Japanese to English, and the experiments also show that vocabulary coverage has a strong correlation with sentiment classification accuracy. Third, translating all the sentiment data in one language into the other language is a time-consuming and labor-intensive job in reality.

In this paper, we propose a deep learning approach, which uses stacked autoencoders [2] to learn language-independent distributed representations of data for cross-lingual sentiment classification. Our model is firstly trained on a large-scale bilingual parallel data and then projects the source language and the target language into a bi-lingual space that fuses the two types of information together. The goal of our model is to learn distributed representations through a hierarchy of network architectures. The learned distributed representations can be used to bridge the gap between the source language and the target language. For example, if we have learned language-independent distributed representations English and Chinese sentiment data, then a classifier trained on labeled English sentiment data can be used to classify Chinese sentiment data.

The novelty of our approach lies in that we employs a deep learning approach to project the source language and the target language into a language-independent unified representations. Our work shares certain intuition with the mixture model for cross-lingual sentiment classification [9] and the bilingual word embeddings used in cross-lingual sentiment classification [11] and phrase-based machine translation [29]. A common property of these approaches is that a word-level alignment (extracted using GIZA++) of bilingual parallel corpus is leveraged [8,9,11,29]. In this paper, we only require alignment parallel sentences and do not rely on word-level alignments of bilingual corpus during training, which simplifies the learning procedure.

To evaluate the effectiveness of the proposed approach, we conduct experiments on the task of English-Chinese cross-lingual sentiment classification. The empirical results show the proposed approach is very effective for cross-lingual sentiment classification, and outperforms other comparison methods.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents our proposed learning distributed semantics for cross-lingual sentiment classification. Section 4 presents the experimental results. Finally, we conclude this paper in section 5.

2 Related Work

2.1 Monolingual Sentiment Classification

Sentiment classification has gained wide interest in natural language processing (NLP) community. Methods for automatically classifying sentiments expressed in products and movie reviews can roughly be divided into supervised and unsupervised (or semi-supervised) sentiment analysis. Supervised techniques have been proved promising and widely used in sentiment classification [13,14,7]. However, the performance of these methods relies on manually labeled training data. In some cases, the labeling work may be time-consuming and expensive. This motivates the problem of learning robust sentiment classification via unsupervised (or semi-supervised) paradigm.

The most representative way to perform semi-supervised paradigm is to employ partial labeled data to guide the sentiment classification [4,18,6]. However, we do not have any labeled data at hand in many situations, which makes the unsupervised paradigm possible. The most representative way to perform unsupervised paradigm is to use a sentiment lexicon to guide the sentiment classification [22,20,28] or learn sentiment orientation of a word from its semantically related words mined from the lexicon [15]. Sentiment polarity of a word is obtained from off-the-shelf sentiment lexicon, the overall sentiment polarity of a document is computed as the summation of sentiment scores of the words in the document. All these work focuses on monolingual sentiment classification, we point the readers to recent books [14,7] for an in-depth survey of literature on sentiment classification.

2.2 Cross-Lingual Sentiment Classification

Cross-lingual sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of data in a label-scare target language by exploiting labeled data from a label-rich language. The fundamental challenge of cross-lingual learning stems from a lack of overlap between the feature spaces of the source language data and that of the target language data.

To bridge the language gap, previous work in the literature mainly relies on machine translation engines or bilingual lexicons to directly adapt labeled data from the source language to the target language. Banea et al. [1] employed the machine translation engines to bridge the language gap in different languages for multilingual subjectivity analysis. Wan [23] and Wan [24] proposed to use ensemble methods to train Chinese sentiment classification model on English labeled data and their Chinese translations. English labeled data are first translated into

Chinese, and then the bi-view sentiment classifiers are trained on English and Chinese labeled data respectively. Pan et al. [12] proposed a bi-view non-negative matrix tri-factorization (BNMTF) model for cross-lingual sentiment classification problem. They employed machine translation engines so that both training and test data are able to have two representations, one in source language and the other in target language. The proposed model is derived from the non-negative matrix factorization models in both languages in order to make more accurate prediction. Prettenhofer and Stein [16] proposed a cross-lingual structural correspondence learning (CL-SCL) method to induce language-independent features. Instead of using machine translation engines to translate labeled text, the authors first selected a subset of pivot features in the source language to translate them into the target language, and then use these pivot pairs to induce cross-lingual representations by modeling the correlations between pivot features and non-pivot features in an unsupervised fashion. Recently, Xiao and Guo [27] used the similar idea with [16] for cross-lingual sentiment classification. Instead of in an fully unsupervised fashion, Xiao and Guo [27] performed representation learning in a semi-supervised manner by directly incorporating discriminative information with respect to the target prediction task. In this paper, we propose a deep learning approach, which uses stacked autoencoders [2] to learn language-independent distributed representations of data instead of machine translation engines.

Another group of works propose to use an unlabeled parallel corpus to induce language-independent representations [8,9]. They assume parallel sentences in the corpus should have the same sentiment polarity and labeled data in both the source and target languages are available. However, this method requires labeled data in both the source and target language, which are not always readily available [9]. Meng et al. [9] proposed a generative cross-lingual mixture model (CLMM) to learn previously unseen sentiment words from the large bilingual parallel data. A common property of this approach is that a word-level alignment (extracted using GIZA++) of bilingual parallel corpus is leveraged [9]. In this paper, we only require alignment parallel sentences and do not rely on word-level alignments of bilingual corpus during training, which simplifies the learning procedure.

3 Learning Distributed Semantics for Cross-Lingual Sentiment Classification

3.1 Model Formulation

Recently, parallel data in multiple languages provides an alternative way for multiview representations, as parallel texts share their semantics, and thus one language can be used to ground the other. Some work has exploited this idea to learn distributed representations at the word level [11,29]. A common property of these approaches is that a word-level alignment (extracted using GIZA++) of bilingual parallel corpus is leveraged [11,29]. In this paper, we only require

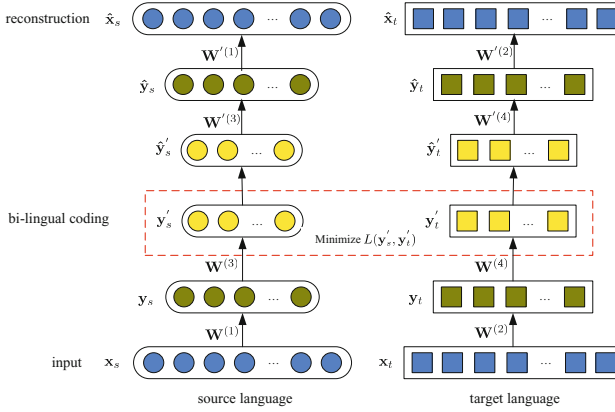


Fig. 1. Denoising stacked autoencoders (DAEs) trained on large-scale parallel sentence pairs $(\mathbf{x}_s, \mathbf{x}_t)$. Input to the model are binary bag-of-words vector representations obtained from the source language and the target language. The model minimize the distance between the sentence level bi-lingual coding of bitext $L(\mathbf{y}'_s, \mathbf{y}'_t)$ as well as the reconstruction errors from the source language and the target language.

alignment parallel sentences and do not rely on word-level alignments of bilingual corpus during training, which simplifies the learning procedure.

Given a large-scale parallel sentence pairs $(\mathbf{x}_s, \mathbf{x}_t)$, we would like to use it to learn distributed representations in both languages that are aligned. The idea is that a shared representation of two parallel sentences would be forced to capture the common information between two languages. Figure 1 shows the model architecture. For each sentence with binary bag-of-words representation \mathbf{x}_s in the source language and an associated binary bag-of-words representation \mathbf{x}_t for the same sentence in the target language, we use the hyperbolic tangent function as the activation function for an encoder f_θ and a decoder $g_{\theta'}$. The weights of each autoencoder are tied, i.e., $\mathbf{W}'^{(1)} = \mathbf{W}^{(1)}$ in Figure 1. We employ denoising stacked autoencoders (DAEs) for pre-training the sentences in each language. For example in Figure 1, let $\tilde{\mathbf{x}}_s$ and denote the corrupted versions of the initial input vector \mathbf{x}_s , we have the following high-level latent representations: $\mathbf{y}_s = f_{\theta_s}(\tilde{\mathbf{x}}_s) = s(\mathbf{W}^{(1)}\tilde{\mathbf{x}}_s + \mathbf{b}^{(1)})$, $\mathbf{y}'_s = f_{\theta_s}(\mathbf{y}_s) = s(\mathbf{W}^{(3)}\mathbf{y}_s + \mathbf{b}^{(3)})$. Essentially, the same steps repeat for the input vector \mathbf{x}_t .

During the decoding phase, we want to be able to perform a reconstruction of the original sentence in any of the languages. In particular, given a representation in any language, we'd like a decoder $g_{\theta'_s}$ that can perform a reconstruction in the source language and another decoder $g_{\theta'_t}$ that can perform a reconstruction in the target language. Given the reconstruction layers, we have $\hat{\mathbf{y}}'_s = g_{\theta'_s}(\mathbf{y}'_s) = s(\mathbf{W}'^{(5)}\mathbf{y}'_s + \mathbf{b}'^{(5)})$, $\hat{\mathbf{y}}_s = g_{\theta'_s}(\hat{\mathbf{y}}'_s) = s(\mathbf{W}'^{(3)}\hat{\mathbf{y}}'_s + \mathbf{b}'^{(3)})$, and $\hat{\mathbf{x}}_s = g'_{\theta'_s}(\hat{\mathbf{y}}_s) = s(\mathbf{W}'^{(1)}\hat{\mathbf{y}}_s + \mathbf{b}'^{(1)})$. Essentially, the same steps repeat for the reconstruction process of $\hat{\mathbf{x}}_t$.

The encoder/decoder decomposition allows us to learn a mapping within each language and across the languages. Specially, for a given parallel sentence pair $(\mathbf{x}_s, \mathbf{x}_t)$, we can train the model to (1) reconstruct \mathbf{x}_s from itself (loss $L(\mathbf{x}_s, \hat{\mathbf{x}}_s)$);

(2) reconstruct \mathbf{x}_t from itself (loss $L(\mathbf{x}_t, \hat{\mathbf{x}}_t)$); and (3) distance between the sentence level encoding of the bitext (loss $L(\mathbf{y}'_s, \mathbf{y}'_t)$). The overall objective function is therefore the weight sum of these errors over a set of binary bag-of-words input vectors $\mathcal{C} = \{(\mathbf{x}_s^{(1)}, \mathbf{x}_t^{(1)}), \dots, (\mathbf{x}_s^{(n)}, \mathbf{x}_t^{(n)})\}$:

$$J(\mathbf{x}_s, \mathbf{x}_t; \theta, \theta') = \sum_{(\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}} \{L(\mathbf{x}_s, \hat{\mathbf{x}}_s) + L(\mathbf{x}_t, \hat{\mathbf{x}}_t) + L(\mathbf{y}'_s, \mathbf{y}'_t) + \frac{\lambda}{2}(\|\theta\|_2 + \|\theta'\|_2)\} \quad (1)$$

where L is a loss function, such as cross-entropy. $\theta = \{\theta_s, \theta_t\}$ and $\theta' = \{\theta'_s, \theta'_t\}$ are the set of all model parameters. Note that we use tied weights for the stacked autoencoder, i.e., $\mathbf{W}^{(1)} = \mathbf{W}'^{(1)}$. In our experiments, we also add the constraints $\mathbf{b}^{(1)} = \mathbf{b}^{(2)}$, $\mathbf{b}^{(3)} = \mathbf{b}^{(4)}$, $\mathbf{b}'^{(1)} = \mathbf{b}'^{(2)}$ and $\mathbf{b}'^{(3)} = \mathbf{b}'^{(4)}$ before the nonlinearity across encoders, to encourage the encoders in both languages to produce representations on the same scale.

3.2 Learning Algorithm

Let $\theta = \{\theta_s, \theta_t\} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{W}^{(4)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \mathbf{b}^{(3)}, \mathbf{b}^{(4)}\}$ and $\theta' = \{\theta'_s, \theta'_t\} = \{\mathbf{W}'^{(1)}, \mathbf{W}'^{(2)}, \mathbf{W}'^{(3)}, \mathbf{W}'^{(4)}, \mathbf{b}'^{(1)}, \mathbf{b}'^{(2)}, \mathbf{b}'^{(3)}, \mathbf{b}'^{(4)}\}$ be the set of our model parameters, then the gradient becomes:

$$\frac{\partial L}{\partial \theta} = \sum_{(\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}} \frac{\partial J(\mathbf{x}_s, \mathbf{x}_t; \theta, \theta')}{\partial \theta} + \lambda \theta. \quad (2)$$

The gradient can be computed efficiently via backpropagation. Since the derivation of the minimization of the distance between the sentence-level bi-lingual coding of bitext and the reconstruction errors can also modify the matrices $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{W}^{(3)}$ and $\mathbf{W}^{(4)}$, the above objective is not necessarily continuous and a step in the gradient descent direction may not necessarily decrease the objective. However, we find that L-BFGS run over the unlabeled parallel data to minimize the objective works well in practice, and that convergence is smooth, with the algorithm typically finding a good solution quickly.

3.3 Cross-Lingual Sentiment Classification

Once we have learned the parameters θ and θ' , we can transform the binary bag-of-words representation of the training data from the source language into the bi-lingual coding space using the learned parameter θ , and then train a simple sentiment classification model using a linear support vector machine (SVM) [5]. For each of the test data from the target language, we also transform its bag-of-words representations into the bi-lingual coding space using the learned parameter θ' and then predict the sentiment polarity of the test data using the trained classification model.

Table 1. Statistics of data sets used in this paper

	MPQA	NTCIR-EN	NTCIR-CH
Positive	1,471 (30%)	528 (30%)	2,378 (55%)
Negative	3,487 (70%)	1,209 (70%)	1,916 (44%)
Total	4,958	1,737	4,294

4 Experiments

4.1 Experimental Setup

In this section, we conduct experiments for cross-lingual sentiment classification. We focus on the two common cross-lingual sentiment classification settings. In the first setting, no labeled data in the target language are available. This task has realistic significance, since in some situations we need to quickly develop a sentiment classifier for languages that we do not have labeled data in hand. In this case, we classify documents in the target language using only labeled data in the source language. In the second setting, we have some labeled data in the target language. In this case, a more reasonable method is to make full use of the labeled data in the source language and the target language to build the sentiment classification model. In our experiments, for each setting, we consider two cases, one is English as the source language and Chinese as the target language, another is Chinese as the source language and English as the target language.

4.2 Data Set

For cross-lingual sentiment classification, we use the benchmark data set described in [8,9]. The labeled data sets consist of two English data sets and one Chinese data set.

MPQA-EN (Labeled English Data): The multi-perspective question answering (MPQA-EN) corpus [25] consists of newswire documents manually labeled with subjectivity information. Following the literature [8], we also discard the sentences with both positive and negative strong expressions.

NTCIR-EN (Labeled English Data) and NTCIR-CH (Labeled Chinese Data): The NTCIR opinion analysis task [19] provides sentiment labeled news data in Chinese and English. The sentences with a sentiment polarity agreed to by at least two annotators are extracted. In this paper, we use the Chinese data from NTCIR-6 as our Chinese labeled data, the English data from NTCIR-6 and NTCIR-7 as our English labeled data. The Chinese sentences are segmented using the Stanford Chinese word segmenter [21].

The statistics of the data sets are shown in Table 1. In our experiments, we evaluate four settings of the data: (1) MPQA-EN \rightarrow NTCIR-CH; (2) NTCIR-EN \rightarrow NTCIR-CH; (3) NTCIR-CH \rightarrow MPQA-EN; and (4) NTCIR-CH \rightarrow NTCIR-EN, where the word before an arrow corresponds with the source language and the word after an arrow corresponds with the target language.

To learn the parameters θ and θ' , we use the Chinese-English parallel corpus [10]. As mentioned earlier, unlike the previous work [8,9,11], we do not use any word alignment between these parallel sentences. Specifically, we segment the Chinese sentences using the Stanford Chinese word segmenter [21] and remove all punctuations from the parallel sentences.

4.3 Model Architecture

Our model has many hyper-parameters, we set these parameters empirically as follows: the source language autoencoder (see Figure 1, left side) and the target language autoencoder (see Figure 1, right side) consist of 1000 hidden units which are then mapped to the second hidden layer with 500 units (the corruption parameter is set to $\nu = 0.5$). The 500 source language and the 500 target language hidden units are fed to a bi-lingual autoencoder containing 500 latent units. We use the model described above and the language-independent representations obtained from the output of the bi-lingual latent layer for the cross-lingual task. Note that some performance gains could be expected if these parameters are optimized on the development set.

4.4 Baseline Methods

In our experiments, we compare our proposed DAEs with the following baseline methods:

SVM: This method learns a SVM classifier for each language given the monolingual labeled data. In this paper, SVM-light [5] is used for all the SVM-related experiments.

MT-SVM: This method employs Google Translate (<http://translate.google.com>) to translate the labeled data from the source language (e.g., English) to the target language (e.g., Chinese) and uses the translated results to train a SVM classifier for the target language.

MT-Cotrain: This method is based on a co-training framework described in [24]. For easy description, we assume that the source language is English while the target language is Chinese. First, two monolingual SVM classifiers are trained on English labeled data and Chinese data translated from English labeled data. Second, the two classifiers make prediction on Chinese unlabeled data and their English translation, respectively. Third, the most confidently predicted English and Chinese documents are added to the training set and the two monolingual SVM classifier are re-trained on the expanded training set. Following the literature [9], we repeat the second and third steps 100 times to obtain the final classifiers.

Joint-Train: This method uses English labeled data and Chinese labeled data to obtain initial parameters for two maximum entropy classifiers, and then conduct

Table 2. Sentiment classification accuracy for Chinese only using English labeled data. Improvements of different methods over baseline MT-SVM are shown in parentheses.

	Method	MPQA-EN \rightarrow NTCIR-CH	NTCIR-EN \rightarrow NTCIR-CH
1	SVM	N/A	N/A
2	MT-SVM	54.33	62.34
3	MT-Cotrain	59.11 (+4.78)	65.13 (+2.79)
4	Joint-Train	N/A	N/A
5	CLMM	71.52 (+17.19)	70.96 (+8.62)
6	DRW	72.27 (+17.94)	71.63 (+9.29)
7	DAEs	72.85 (+18.52)	72.21 (+9.87)

EM-iterations to update the parameters to gradually improve the agreement of the two monolingual classifiers on the unlabeled parallel data [8].

CLMM: This method proposes a generative cross-lingual mixture model (CLMM) [9] and learns previously unseen sentiment words from the large-scale bilingual parallel data to improve the vocabulary coverage.

DRW: This is the state-of-the-art method for cross-lingual sentiment classification described in [11]. This method learns distributed representations of words via multitask and word alignment for cross-lingual sentiment classification.

4.5 Cross-Lingual Sentiment Classification Only Using Source Language Labeled Data

In this section, we investigate cross-lingual sentiment classification towards the case that we have only labeled data from the source language. The first set of experiments are conducted on using only English labeled data to build sentiment classifier for Chinese sentiment classification. This is a challenging task since we do not have any Chinese labeled data in hand.

Table 2 shows the accuracy of the baseline systems as well as the proposed model (DAEs). As seen from the table, our proposed approach DAEs outperforms all baseline methods for Chinese sentiment classification only using the labeled English data. Specifically, our proposed approach improves the accuracy, compared to MT-SVM, by 18.52% and 9.87% (row 2 vs. row 7) on Chinese in the first setting and in the second setting, respectively. Meanwhile, the accuracy of MT-SVM on NTCIR-EN \rightarrow NTCIR-CH is much better than that on MPQA-EN \rightarrow NTCIR-CH. The reason may be that NTCIR-EN and NTCIR-CH cover similar topics. Besides, we also observe that using a parallel corpus instead of machine translations can improve the classification accuracy (row 2 and row 3 vs. row 5, row 6 and row 7). Moreover, Our proposed DAEs outperforms CLMM and DRW (row 5 and row 6 vs. row 7, the comparisons are mildly significant with t -test (p -value < 0.08)). The reason may be that our method can effectively learn sentence-level distributed representations rather than using the off-the-shelf word alignment tools (e.g., GIZA++) to bridge the language gap.

Table 3. Sentiment classification accuracy for English only using Chinese labeled data. Improvements of different methods over baseline MT-SVM are shown in parentheses.

	Method	NTCIR-CH \rightarrow MPQA-EN	NTCIR-CH \rightarrow NTCIR-EN
1	SVM	N/A	N/A
2	MT-SVM	52.47	58.51
3	MT-Cotrain	58.63 (+6.16)	63.72 (+5.21)
4	Joint-Train	N/A	N/A
5	CLMM	68.29 (+15.82)	69.15 (+10.64)
6	DRW	70.85 (+18.38)	72.57 (+14.06)
7	DAEs	71.42 (+18.95)	73.38 (+14.87)

Table 4. Sentiment classification accuracy for Chinese by using English and Chinese labeled data. Improvements of different methods over baseline SVM are shown in parentheses.

	Method	MPQA-EN \rightarrow NTCIR-CH	NTCIR-EN \rightarrow NTCIR-CH
1	SVM	80.58	80.58
2	MT-SVM	54.33 (-26.25)	62.34 (-18.24)
3	MT-Cotrain	80.93 (+0.35)	82.28 (+2.79)
4	Joint-Train	83.42 (+2.84)	83.11 (+2.53)
5	CLMM	83.02 (+2.44)	82.73 (+2.15)
6	DRW	83.54 (+2.96)	83.26 (+2.68)
7	DAEs	83.81 (+3.23)	83.59 (+3.01)

The second set of experiments are conducted on using only Chinese labeled data to build sentiment classifier for English sentiment classification. Table 3 shows the sentiment classification accuracy for English using only Chinese labeled data. From this table, we have the similar observations as in Table 2.

4.6 Cross-Lingual Sentiment Classification Using Source Language and Target Language Labeled Data

The third set of experiments are conducted on using both English labeled data and Chinese labeled data to build the Chinese sentiment classifier. We conduct 5-fold cross validation on Chinese labeled data and use the similar settings with [9].

Table 4 shows the average accuracy of baseline systems as well as our proposed DAEs. From this table, we can see that SVM performs significantly better than MT-SVM. The reason may be that we use the original Chinese labeled data instead of translated Chinese labeled data. We also find that all four methods which employ the unlabeled parallel corpus, namely MT-Cotrain, Joint-Train, CLMM and DAEs, still show improvements over the baseline SVM. Moreover, our proposed DAEs outperforms than DRW and obtains the state-of-the-art accuracy on both data sets. This again validates that learning sentence-level distributed representations is better than using word alignment tools for cross-lingual sentiment classification. Due to limited space, we do not present the

experimental results for English and some other related discussions, we will leave these works for further research.

5 Conclusion

In this paper, we present a model that uses stacked autoencoders to learn distributed representations through a hierarchy of network architectures. The learned distributed representations can be used to bridge the gap between the source language and the target language. To evaluate the effectiveness of the proposed approach, we conduct experiments on the task of English-Chinese cross-lingual sentiment classification. The empirical results show the proposed approach is effective for cross-lingual sentiment classification, and outperforms other comparison methods.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 61303180, No. 61272332 and No. 61333018), the Beijing Natural Science Foundation (No. 4144087), CCF Opening Project of Chinese Information Processing, and also Sponsored by CCF-Tencent Open Research Fund. We thank the anonymous reviewers for their insightful comments.

References

1. Banea, C., Mihalcea, R., Wiebe, J., Hassan, S.: Multilingual Subjectivity Analysis Using Machine Translation. In: Proceedings of EMNLP, pp. 127–135
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Proceedings of NIPS. Universite De Montreal, Montreal Quebec
3. Duh, K., Fujino, A., Nagata, M.: Is Machine Translation Ripe for Cross-lingual Sentiment Classification? In: Proceedings of ACL, pp. 429–433
4. Goldberg, A.B., Zhu, X.: Seeing Stars when There Aren'T Many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization. In: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, pp. 45–52
5. Joachims, T.: Making Large-scale Support Vector Machine Learning Practical. *Advances in Kernel Methods*, pp. 169-184
6. Li, S., Wang, Z., Zhou, G., Lee, S.Y.M.: Semi-supervised Learning for Imbalanced Sentiment Classification. In: Proceedings of IJCAI, pp. 1826–1831
7. Liu, B.: Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*
8. Lu, B., Tan, C., Cardie, C., Tsou, B.K.: Joint Bilingual Sentiment Classification with Unlabeled Parallel Corpora. In: Proceedings of ACL, pp. 320–330
9. Meng, X., Wei, F., Liu, X., Zhou, M., Xu, G., Wang, H.: Cross-lingual Mixture Model for Sentiment Classification. In: Proceedings of ACL, pp. 572–581
10. Munteanu, D.S., Marcu, D.: Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Comput. Linguist.*, 477–504
11. Klementiev, A., Titov, I., Bhattarai, B.: Inducing Crosslingual Distributed Representations of Words. In: Proceedings of COLING

12. Pan, J., Xue, G.-R., Yu, Y., Wang, Y.: Cross-lingual sentiment classification via bi-view non-negative matrix tri-factorization. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 289–300. Springer, Heidelberg (2011)
13. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In: Proceedings of EMNLP, pp. 79–86
14. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. In: Found. Trends Inf. Retr., pp. 1–135
15. Peng, W., Park, D.H.: Generate Adjective Sentiment Dictionary for Social Media Sentiment Analysis Using Constrained Nonnegative Matrix Factorization. In: Proceedings of ICWSM
16. Prettenhofer, P., Stein, B.: Cross-language Text Classification Using Structural Correspondence Learning. In: Proceedings of ACL, pp. 1118–1127
17. Shanahan, J.G., Grefenstette, G., Qu, Y., Evans, D.A.: Mining Multilingual Opinions through Classification and Translation. In: Proceedings of CIKM
18. Sindhvani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: Proceedings of ICDM
19. Seki, Y., Evans, D.K., Ku, L.-W., Chen, H.-H., Kando, N., Lin, C.-Y.: Overview of Opinion Analysis Pilot Task at NTCIR-6. In: Proceedings of the Workshop Meeting of the National Institute of Informatics (NII) Test Collection for Information Retrieval Systems (NTCIR), pp. 265–278
20. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based Methods for Sentiment Analysis. *Comput. Linguist.* 267-307
21. Tseng, H.: A conditional random field word segmenter. In: Fourth SIGHAN Workshop on Chinese Language Processing
22. Turney, P.D.: Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceedings of ACL, pp. 417–424
23. Wan, X.: Using Bilingual Knowledge and Ensemble Techniques for Unsupervised Chinese Sentiment Analysis. In: Proceedings of EMNLP, pp. 553–561
24. Wan, X.: Co-training for Cross-lingual Sentiment Classification. In: Proceedings of ACL-IJCNLP, pp. 235–243
25. Wiebe, J., Cardie, C.: Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation. Language Resources and Evaluation (formerly Computers and the Humanities)*
26. Wu, K., Wang, X., Lu, B.-L.: Cross language text categorization using a bilingual lexicon. In: Proceedings of IJCNLP
27. Xiao, M., Guo, Y.: Semi-Supervised Representation Learning for Cross-Lingual Text Classification. In: Proceedings of ACL, pp. 1465–1475
28. Zhou, G., Zhao, J., Zeng, D.: Sentiment Classification with Graph Co-Regularization. In: Proceedings of COLING, pp. 1331–1340
29. Zou, W.Y., Socher, R., Cer, D.M., Manning, C.D.: Bilingual Word Embeddings for Phrase-Based Machine Translation. In: Proceedings of ACL, pp. 1393–1398

A Short Texts Matching Method Using Shallow Features and Deep Features

Longbiao Kang¹, Baotian Hu¹, Xiangping Wu¹, Qingcai Chen^{1,*}, and Yan He²

¹ Intelligent Computing Research Center,
School of Computer Science and Technology,
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
{klbgx7,baotianchina,wxpleduole,qingcai.chen}@gmail.com

² Zunyi Medical and Pharmaceutical College, Zunyi, China
xheyan@21cn.com

Abstract. Semantic matching is widely used in many natural language processing tasks. In this paper, we focus on the semantic matching between short texts and design a model to generate deep features, which describe the semantic relevance between short “text object”. Furthermore, we design a method to combine shallow features of short texts (i.e., LSI, VSM and some other handcraft features) with deep features of short texts (i.e., word embedding matching of short text). Finally, a ranking model (i.e., RankSVM) is used to make the final judgment. In order to evaluate our method, we implement our method on the task of matching posts and responses. Results of experiments show that our method achieves the state-of-the-art performance by using shallow features and deep features.

Keywords: Short Text, Semantic Matching, Word Embedding, Ranking Model.

1 Introduction

Many natural language processing (NLP) tasks (such as, paraphrase identification [11], information retrieval [8]) can be reduced to semantic matching problems. In this paper, we focus on a short text-matching task called short text conversation, firstly defined by Wang et.al [3]. For a given short text such as a post, this task aims to find a massive suitable response from the candidate set. It’s a simplified task of modeling a complete dialogue session such as Turing test. For the convenience of description, we give the following example, the post P (post), $R+$ (positive response), $R1-$ (negative response) and $R2-$ (negative response).

- P : 深圳 今天 天气 怎么样? *How is the weather like today in Shenzhen?*
- $R+$: 深圳 现在 正 大雨磅礴。 *It's pouring down in torrents now in Shenzhen.*

* Corresponding author.

- R1-: 在深圳过的怎么样？*How is everything going in Shenzhen?*
- R2-: 今天么海天气不错哦。*The weather in Shanghai is very good today.*

The semantic matching is a challenging problem, since it aims to find the semantic relevance between two “text objects”. Wang et al. [3] describe a retrieval-based model, which uses about 15 shallow matching features. Most of the features are learned from matching models or generated directly from posts and responses. Although the retrieval-based model performs reasonably well on the post-response matching task, it can only capture the word-by-word matching and latent semantic matching features between posts and responses. For retrieval-based model, it is difficult to recognize R1- as a negative response. However, it is easy to recognize R2- as a negative response, only if we add a feature describing whether the named entities from the post and response are the same. Distributed representation (also called word embedding) of text induced from deep learning is well suited for this task, because it contains rich semantic information of text and can model the relevance between different words or phrases. Related works have demonstrated that the embedding-based method can capture rich semantic relevance between “text objects” and performs well on tasks like paraphrase identification [11] and information retrieval [8]. However, they are not powerful enough at handling the subtlety for specific task. For embeddings of “上海”(Shanghai) and “深圳”(Shenzhen) are very close, it is difficult for embedding-based method to recognize R2- as negative response to P.

In this paper, we study the shallow features and deep features for short text matching and try to combine them to improve the performance. The remainder of this paper is organized as follows. Section 2 reviews the relevant works for our task. Section 3 describes our model for this task. Experimental design, comparison and analysis are presented in Section 4. Finally, we make conclusions in the Section 5.

2 Related Works

Previous works often use rule-based or learning-based models for modeling a complete dialogue [4, 5, 13]. These methods require well-designed rules or particular learning algorithms but relatively less training data. Recent years, by leveraging the massive short text data collected by social media and information retrieval techniques, researchers attack this problem from a new angle [2, 10]. Wang, et al. [3] released a short text dataset collected from Sina Weibo and proposed a retrieval-based response model for short-text conversation. This model considers semantic matching between posts and responses, then retrieves and returns the most appropriate response for a given post. Most of the features used in this model are latent semantic features, which can’t capture the deep semantic relevance between posts and responses. In this paper we start from this dataset and combine shallow features and deep features to improve the performance of the short-text conversation.

Deep learning is another approach to solve this task. Such works in that thread include deep architecture for matching short texts (DeepMatch) proposed by Lu and Li [16] and deep structured semantic models for information retrieval proposed by

Huang et al. [8]. Lu and Li [16] use a deep architecture to combine the localness and hierarchy intrinsic to natural language problem by using a massive dataset. However, this architecture almost use the traditional features instead of deep features of short text (i.e., word embedding). Most of these works, which claimed to have used deep architecture, are embedding-based models. Word embedding, also called distributed representation of words [1, 6, 9, 15], shows strong power for measuring syntactic and semantic word similarities and has achieved success in many NLP tasks such as sentiment analysis [12] and statistical machine translation [7]. Our work is also related with Socher et al. [11], which use the unfolding recursive auto-encoder and parsing tree to construct the interaction matrix of two sentences. This strategy combining with dynamic pooling achieves state-of-the-art performance on paraphrase identification task.

3 Matching Short Texts Using Both Shallow and Deep Features

The previous work [3] learned a ranking model with 15 shallow matching features, which are learned from matching models or generated directly from post and response. In our method, we build three new matching models and learn both shallow features and deep features between posts and responses. Then we build a ranking model to evaluate the candidate responses with both shallow features and deep features.

For shallow matching features, we first introduce a variant of vector space model to measure post-response similarity by a word-by-word matching. Then we use LSI (Latent semantic indexing) model to capture the latent semantic matching features, which may not be well captured by a word-by-word matching. For deep matching features, we construct a matching models based on neural networks and word embedding.

3.1 Shallow Matching Features

Post-Response Similarity: To measuring the similarity between a post and a response by a word-by-word matching, we use a simple vector space model. For example, given a post and a response, the sparse representations of them in vector space model are x and y , then the Post-Response Similarity can be measured by the cosine similarity.

$$\text{Sim}(x, y) = \text{CosineSim}(x, y) = \frac{x^T y}{\|x\| \|y\|}$$

Unlike the traditional vector space model such as Bag-of-words model, in our model the values in the vector are the TF-IDF weights of words. Vector space model based on TF-IDF weights is valuable words bias and shows better performance than traditional Bag-of-words model in the experiment.

Post-Response Latent Semantic Similarity: LSI model has been used in many NLP task for measuring latent semantic matching. It learns a mapping from the original sparse representation to a low-dimensional and dense representation for text. For example, we represent post as vector x and response as vector y . Using LSI model, we map x and y to low-dimensional vector representations x_{lsi} and y_{lsi} . Similarly, the semantic similarity between the post and the response can be measured by the cosine similarity of x_{lsi} and y_{lsi} .

$$\text{SemSim}_{PR}(x_{lsi}, y_{lsi}) = \text{CosineSim}(x_{lsi}, y_{lsi})$$

Post-Post Latent Semantic Similarity: Inspired by previous work [3], we also consider the semantic similarity between a post and a post. The intuition is that a response y is suitable for a post x if its original post x' is similar to x . So we use semantic similarity between x and the original post x' to measure this kind of indirect correlation between a post and a response. Here we also use the LSI model to measure the post-post semantic similarity.

$$\text{Cor}(x, y) = \text{SemSim}_{PP}(x_{lsi}, x'_{lsi}) = \text{CosineSim}(x_{lsi}, x'_{lsi})$$

3.2 Deep Matching Feature

As is stated above, we can acquire shallow matching features between a post and a response. To learn the deep matching features between posts and responses, we propose a matching model based on neural networks and word embedding.

With the learned word embeddings, we first map every word of posts and responses to a unique vector. As shown below, given a post x or a response y , we first convert them into a set of vectors.

$$\begin{aligned} x &= (w_1 w_2 w_3 \dots w_i \dots) \rightarrow X = (v_1 v_2 v_3 \dots v_i \dots) \\ y &= (w_1 w_2 w_3 \dots w_j \dots) \rightarrow Y = (v_1 v_2 v_3 \dots v_j \dots) \end{aligned}$$

Here w_i is the i -th word in x and v_i is the corresponding word embedding. By measuring cosine similarity for each vector pair in X and Y , we can get a correlation matrix M_{cor} .

$$M_{cor} = \left(\text{CosineSim}(v_i, v_j) \right)_{v_i \in X, v_j \in Y}$$

The size of this correlation matrix is variable for variable-length posts and responses. In order to use neural networks and prevent the dimension of the feature space becoming too large, we need to convert variable-length posts and responses to fixed-length. We sort all the words in post x and response y by their TF-IDF weights in descending order, then we choose the top m words in x and top n words in y . Finally, we can get a correlation matrix with size $m*n$.

$$\text{sorted}(x) \xrightarrow{\text{top } m \text{ words}} x' = (w_1 w_2 w_3 \dots w_m) \rightarrow X' = (v_1 v_2 v_3 \dots v_m)$$

$$sorted(y) \xrightarrow{\text{top } n \text{ words}} y' = (w_1 w_2 w_3 \dots w_n) \rightarrow Y' = (v_1 v_2 v_3 \dots v_n)$$

$$M_{cor}' = \left(\text{CosineSim}(v_i, v_j) \right) v_i \in X', v_j \in Y'$$

Here X' , Y' and M_{cor}' have fixed size. For the post with a length less than m or the response with a length less than n , we set zero for the corresponding value in the correlation matrix. At the last step of the model, we flatten M_{cor}' to a feature vector with the same size $m*n$. Then we build neural networks with single hidden layer, which uses values of the feature vector as input features and outputs a matching score. The whole model is shown in Figure 1.

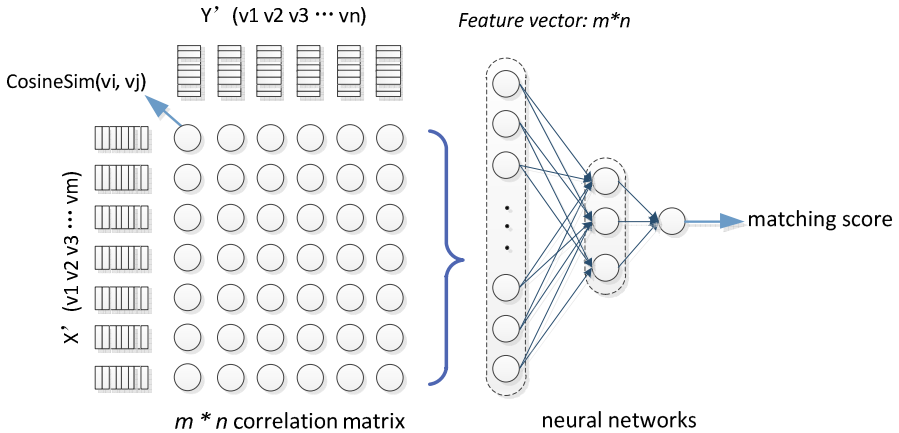


Fig. 1. Matching model based on neural networks and word embedding

Training: For training this model, we use the ranking-based strategy. Given a post x , the model should output a higher score for positive response $y+$ than negative response $y-$. So our instance for training is $(x, y+, y-)$. We use the unlabeled data to train our embedding matching model. For a post x , every original response of x is used as positive response $y+$, and choose a response randomly from the response dataset as negative response $y-$ for each $y+$. Hence, we get the following ranking-based loss as objective:

$$E_{\theta}(x, y^+, y^-) = \max(0, \alpha + s(x, y^-) - s(x, y^+))$$

Where $s(x, y)$ is the output matching score for (x, y) , α is the margin between positive response and negative response, θ is the parameters for the embedding match model. The optimization is relatively straightforward with the back-propagation.

3.3 Other Matching Features

In addition to the matching features generated from the matching models above, we also use some handcraft features for this task, which can describe the relevance between post and response for some special cases.

- $Common_Entity(x, y)$: This feature measures whether post x and response y have same entity words such as first and last names, geographic locations, addresses, companies and organization name. The intuition here is that a post and a response in the nature conversation usually contain some common key words.
- $Common_Number(x, y)$: This feature measures whether post x and response y have same number such as date and money.
- $Common_Words_Lenght(x, y)$: This feature indicates the length of the longest common string between a post and a response. In social media such as Micro-blog, a response can be forwarded as a new post. We may find a response which is very similar to the given post, but it's not a suitable response. With this feature, we can filter this kind of responses.
- $Length_Ratio(x, y)$: This feature indicates the ratio of the length of post x to the length of response y .

3.4 Ranking and Combination of Features

At the last step of our method, a ranking model with all the matching features above is learned to further evaluate the candidate responses. As shown in Figure 2, given a post, the ranking model gives a matching score for each candidate response. Then we pick the response with the highest matching score from the candidate set as suitable response for the post.

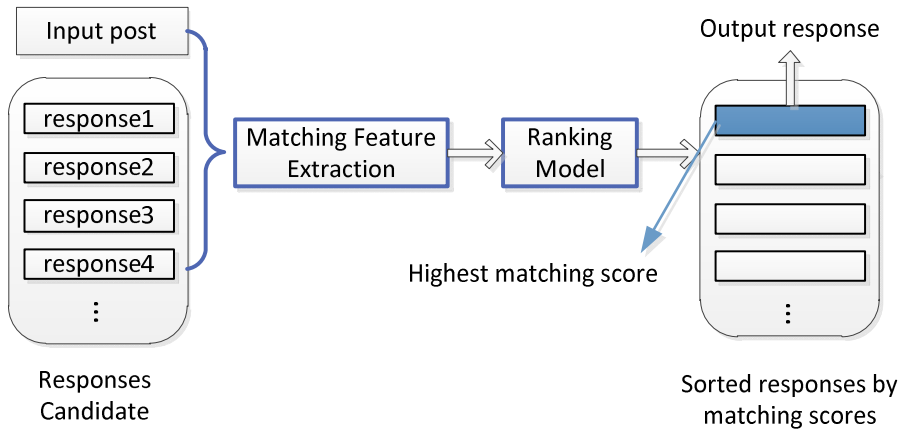


Fig. 2. Ranking model for short texts matching

The ranking function of the ranking model is defined as following, which is a linear score function and trained with RankSVM [14].

$$Score(x, y) = \sum_{i=1}^n w_i \Phi_i(x, y)$$

Here $\Phi_i(x, y)$ stands for the acquired matching features and w_i is the weight of $\Phi_i(x, y)$ to be learned.

4 Experiment

4.1 Dataset

In our experiments, we made use of the dataset of short-text conversation based on the real-world instances from Sina Weibo, which is published by Wang, et al. [3], as shown in Table 1.

Table 1. Dataset of short-text conversation based on the real-world instances from Sina Weibo

Dataset	Size
Unlabeled post-response pairs	38,016 posts, 618,104 responses
Labeled post-response pairs	422 posts, 12,402 responses
Vocabulary	125,817

This dataset includes two parts: unlabeled post-response pairs and labeled post-response pairs. The unlabeled post-response pairs are training set, which contains 38,016 posts and 618,104 responses. The labeled post-response pairs are test dataset, which contains 422 posts and 12,402 responses, and there are about 30 candidate responses for each post. The vocabulary of this dataset contains 125,817 words.

For each post-response pair, a post and a response are represented as *post_id* and *response_id*. In the labeled post-response pairs, each pair is labeled as a matched pair (marked as 2) or a mismatched pair (marked as 1) by human. An example of unlabeled post-response pair is shown below:

Post-response pair: 10:81,213

- *Post: 10##2012 年 来 了 ， 祝 好 运 、 健 康 、 佳 肴 伴 你 度 过 一 个 快 乐 新 年 。*
- *Post: 10##2012 is coming. Good luck, good health, hood cheer. I wish you a happy New Year.*
- *Response 1: 81##林 老 师 ， 新 年 快 乐 。*
- *Response 1: 81##Teacher Lin, happy new year!*
- *Response 2: 213##祝 教 授 ， 工 作 顺 利 ， 身 体 健 康 ！*
- *Response 2: 213##Happy new year, good health, professor.*

In order to train word embedding, we also build a Weibo dataset of 33,405,212 posts. We filter out the posts with length less than 5 and the meaningless posts in the dataset. Jieba¹, a famous open source software, is used for word segmentation. After preprocessing and word segmentation, we use the method introduced by Mikolov et al. [6] for learning word embedding. In our experiments, we use a

¹ <https://github.com/fxsjy/jieba>

particular implementation of this model² and the length of word embedding is set to 100. After training, we learned a set of word embeddings with a vocabulary of size 810,214.

4.2 Experiments and Benchmark

To training matching models in Section 3, we use unlabeled post-response pairs introduced in the dataset. For vector space model based on TF-IDF, the TF-IDF values for words can be generated directly from corpus. For training LSI model, we concatenate each post and its responses to get informative documents. For matching model based on word embedding, we need to sort all the words in a post and a response by their TF-IDF weights in descending order, and choose the top m words in the post and top n words in the response for generating matching features. By using comparison experiments, we can get best effect with 15 for m and 10 for n .

After training matching models and generating matching features, we train the ranking model defined in Equation 1 with the implementation of RankSVM³. Specifically we perform a 5-fold cross-validation on the labeled post-response pairs.

Our model will return the response with the highest matching score. So the evaluation of our models is based on the **P@1**. This measures the precision of the response returned by model.

$$P@1 = \frac{\text{Count}(\text{top 1 response is matched})}{\text{Count}(\text{posts})}$$

4.3 Performance

The results of experiments are shown in Table 2. Our competitor methods include retrieval-based model [3] and DeepMatch [16] model. For the DeepMatch model, we re-implement it and train it on the unlabeled training dataset. VSM stands for the matching features generated by vector space model based on TF-IDF. LSI stands for the matching features generated by LSI model. The deep feature is generated by embedding match model.

Table 2. Comparison of different features

Model and Features	P@1
Retrieval-based Response Model [3]	0.574
DeepMatch [16]	0.424
Shallow Features (VSM, LSI)	0.577
Shallow Features + Deep Features (VSM, LSI, Embedding Match)	0.612
All Features	0.637

² <http://radimrehurek.com/2014/02/word2vec-tutorial/>

³ http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html

From the results, we can see that just using the word features generated by VSM model and latent semantic model, the result is as well as retrieval-based model. When combined with deep features, the performance significantly outperforms retrieval-based model and those just using shallow features. This demonstrates that although the shallow features are effective for short text semantic matching, it can't capture the deep semantic relevance information between two text object'. After adding other handcraft features the performance can be improved further, it implies that deep features cannot cover some relevance for some special cases, for example the “上海 天气”(The weather in Shanghai) and“海圳 天气”(The weather in Shenzhen) case. The main reason for the bad performance of DeepMatch may be that the training dataset is not big enough for this deep architecture.

5 Conclusion

In this paper, we design an embedding match model to generate deep features for short text matching and study the effect of shallow features and deep features on the performance. Experiments show that the deep features cover rich semantic relevance information between post and response, which the shallow features cannot capture. Nonetheless, experiment also shows that shallow features are necessary for some special cases of semantic matching. Combining the shallow features and deep feature generated by embedding match model, we get the state-of-the-art performance on the dataset released by Wang et al. [3].

Although the deep feature has proved significantly effective in this paper, there is still much matching information, which it cannot capture. For future work, we will try to use deep architecture and massive data to extract rich deep features of short text matching.

Acknowledgement. This paper is supported in part by grants: NSFCs (National Natural Science Foundation of China) (61173075 and 61272383), Strategic Emerging Industry Development Special Funds of Shenzhen (ZDSY20120613125401420 and JCYJ20120613151940045), Key Basic Research Foundation of Shenzhen (JC201005260118A and JC201005260175A) and Science and Technology Funds of Guizhou Province (黔科合J字[2013]2335).

References

1. Mnih, A., Hinton, G.: Three new graphical models for statistical language modelling. In: International Conference on Machine Learning, ICML (2007)
2. Leuski, A., Traum, D.R.: Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine* 32(2), 42–56 (2011)
3. Wang, H., Lu, Z., Li, H., Chen, E.: A dataset for research on short-text conversations. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, pp. 935–945 (2013)

4. Williams, J.D., Young, S.: Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.* 21(2), 393–422 (2007)
5. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowl. Eng. Rev.*, 97–126 (2006)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
7. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168 (2013)
8. Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, pp. 2333–2338. ACM (2013)
9. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research (JMLR)* 12, 2493–2537 (2011)
10. Jafarpour, S., Burges, C.J.C.: Filter, rank, and transfer the knowledge: Learning to chat (2010)
11. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: *Advances in Neural Information Processing Systems* (2011)
12. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D.: Semisupervised recursive autoencoders for predicting sentiment distributions. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2011)
13. Misu, T., Georgila, K., Leuski, A., Traum, D.: Reinforcement learning of question-answering dialogue policies for virtual museum guides. In: *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2012*, pp. 84–93 (2012)
14. Joachims, T.: Optimizing search engines using clickthrough data. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002*, pp. 133–142. ACM, New York (2002)
15. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)* 3, 1137–1155 (2003)
16. Lu, Z., Li, H.: A deep architecture for matching short texts. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 26, pp. 1367–1375. Curran Associates, Inc. (2013)

A Feature Extraction Method Based on Word Embedding for Word Similarity Computing

Weitai Zhang, Weiran Xu, Guang Chen, and Jun Guo

Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract. In this paper, we introduce a new NLP task similar to word expansion task or word similarity task, which can discover words sharing the same semantic components (feature sub-space) with seed words. We also propose a Feature Extraction method based on Word Embeddings for this problem. We train word embeddings using state-of-the-art methods like word2vec and models supplied by Stanford NLP Group. Prior Statistical Knowledge and Negative Sampling are proposed and utilized to help extract the Feature Sub-Space. We evaluate our model on WordNet synonym dictionary dataset and compare it to word2vec on synonymy mining and word similarity computing task, showing that our method outperforms other models or methods and can significantly help improve language understanding.

Keywords: word embeddings, feature sub-space, negative sampling, word similarity, prior statistical knowledge.

1 Introduction

Word similarity in NLP is a task of computing the similarity between two or more words by certain methods. In general, similarity always means the semantic similarity of words, for example, “apple” and “pear” have very close relationship, while “mike” and “class” are not relevant. Researchers improved the performance of word similarity task within methods like brown cluster, topic model, vector space model and so on[1][2]. Recently, word representation, mostly word embedding, is proved to be excellent at word similarity task [3].

In this paper, we focus on a new NLP task similar to word expansion task or word similarity task which could reveal words having the same semantic components with given seed words. The differentia between this task and word similarity task lies in that (1) it reveals words through more than one word and (2) the key point is how to represent the same semantic components of the seed words. We propose a method combining words’ syntactic information gained with state-of-the-art methods and representation of the same semantic components of the seed words based on word embeddings.

Most words have more than one meaning, which leads to that certain aspects of some words may share the same semantic information. One specific example given here is that “Beijing”, “Shanghai” and “Tokyo” have the same facet that they are all

city names. Another example is that “sad”, “sorrow” and “low” also have a same meaning of sad or upset in mood. On condition of research needs and actual situations, we always need to reveal more words having the same semantic components with the given seed words which is exactly our problem. Like, given “Beijing”, “Shanghai” and “Tokyo”, we can find that “Guangzhou”, “Houston”, “Osaka” are also city names; given “sad”, “sorrow” and “low”, we may find that “upset”, “depressed” also have the meaning of sad or upset.

The structure of this paper is as follows. Section 2 describes Prior Statistical Knowledge proposed. Section 3 describes our method and gives structure of our model. Experimental results are presented in Section 4. Finally, conclusions are made in the last section.

2 Prior Statistical Knowledge

Researchers of deep learning in natural language processing believe word embeddings can represent syntactic information or semantic information [9]. However, current progress shows that word embeddings or neural language model may not outperform state-of-the-art methods in some tasks. For example, Brown Cluster is superior to the word embeddings on NER task [11] and there is only a small difference between Brown clusters and word embeddings on chunking task.

The effectiveness of word representation plays a significant role in our model. To more precisely represent a word, we propose Prior Statistical Knowledge of words to enrich the representation of words and compute using the published open source tools from Stanford NLP Group[10].

Table 1. Samples of labels and features in Prior Statistical Knowledge

Label name	No.	Feature name
POS	29	vb, cc, jjs, prp, in, nnp...
NER	3	person, location, organization
Parsing	94	cc_post, tmod_post, prt_pre, cop_pre ...

As showed in table 1, we assign each word with 3 most important labels: Named Entity Recognition, Part-Of-Speech tagging, Parsing Dependencies. According to Stanford CoreNLP tools and actual situation, we choose 29 features for POS and 3 features for NER. Particularly in Parsing Dependencies part, there are 47 kinds of ternary relation pair in total, so we extract 94 features for Parsing Dependencies, doubled because of the position each word exists. In Parsing Dependencies, features ended with ‘_pre’ indicate that words are in the front of ternary relations and ‘_post’ corresponds to the back of ternary relations. For example, in ternary relation

“subj(undesirable-10,state-9)”, the number of 9 and 10 is the position of words in the sentence; in our method, we assign feature “subj_pre” to “undesirable” and “subj_post” to “state”.

For each label, a word may have several features. For example, as shown in table 2, word “bush” has several features for each label, like “nnp”, “nn”, “jj”, “vb” for POS. To compute the weight of each feature, we utilize the Wikipedia corpus crawled from Wikipedia website. The corpuses are all standard, clean, grammatical articles. We compute the occurrence times of each feature and assign a probability to each feature for each word concluded by normalization. For instance, specific to NER, we assign 0.9748 to “Person” feature because “bush” appears in our corpus with a 0.9748 probability of “Person” according to our statistic results.

We compute the probability of one word with the equation (1).

$$P_{f_i} = \frac{c_i}{\sum_i c_i} \quad (1)$$

where P_{f_i} stands for the weight of feature f_i and c_i stands for the occurrence of feature f_i .

Prior Statistical Knowledge firstly helps reduce calculation amount and improve general speed twice the original model by reducing words number and abandoning useless words of calculation. It can also enrich the representations of words by synthesizing word embeddings and state-of-the-art methods before.

Table 2. Example of a specific feature assignment and feature weight for word “bush”

Label name	Feature name & weight		
POS	nnp	nn	jj
	0.8919	0.0630	0.0445
NER	Person	Organization	Location
	0.9748	0.0190	0.0062
Parsing	pre_post	det_pre	amod_post
	0.2056	0.1772	0.1203

3 Feature Sub-Space Extraction Model

3.1 Feature Sub-Space

As stated before, a word embedding is always a vector associated with each word. Each dimension or several dimensions correspond to a feature and might even have a semantic or grammatical interpretation. Given several seed words that have one or more identical meanings, we believe several common dimensions of these vectors represent the common part of these words although we cannot point out what each

dimension exactly represent. We believe this feature sub-space with a lower dimensionality will reveal and represent the common latent semantic component information.

On the condition that dimensions of vectors are assumed to be identical and independent distribution, we can assume that the values of all words in certain dimension to be Gaussian Distribution if the values are random variables which is always reasonable and effective as a assumption in natural language processing tasks. According to that, values of dimensions sharing the common meaning of the seed words should be Gaussian Distribution and around mean value while values of other dimensions should be very different and away from the mean value. Based on this theory, we propose the feature sub-space extraction model and reveal the common dimensions of seed words.

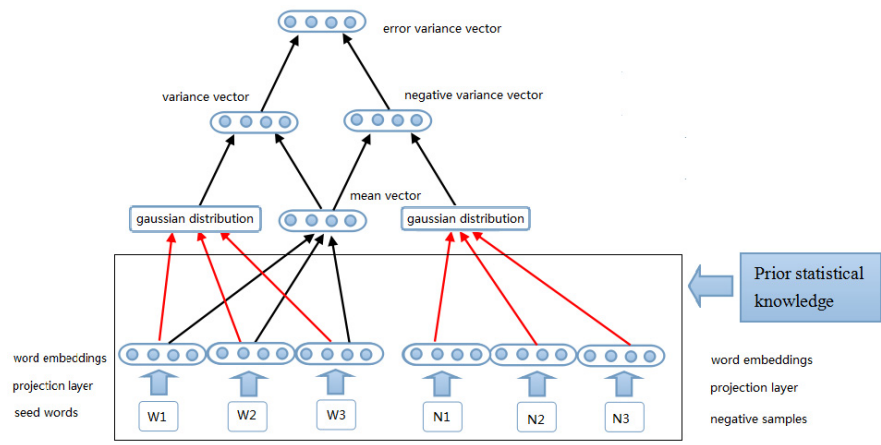


Fig. 1. Feature Sub-Space Extraction Model

3.2 Negative Sampling

Noise Contrastive Estimation (NCE), introduced in [11], posits that a good model should be able to differentiate data from noise by means of logistic regression. In our model, we concerned with extracting high-related sub-space, which means variances of words having the same sub-semantic information with the seed words are small, while those of negative samples are large. So we can simplify NCE as long as the feature sub-space retains its quality.

The main difference between the Negative sampling and NCE is that NCE needs both samples and the numerical probabilities of the noise distribution, while Negative sampling uses only samples [12]. In this paper, we introduce negative sampling because (1) negative samples will help avoid acquiring uncorrelated dimensions where the variances are small enough by coincidence and (2) help enhance the noise immunity of dimensions that are really related to the sub-space.

Negative sampling plays a decisive role in the Feature Sub-Space Extraction Model. After calculating weight of dimensions of seed words and that of negative samples, we extract low-valued dimensions of error vector which is the difference value of \vec{D} and \vec{D}_{neg} .

As showed in Figure I, our model consists of the following steps:

- Utilize Prior Statistical Knowledge and reduce the computing subset. We only consider words with the same features as the seed words in the following steps.
- Map the seed words into word embeddings using lookup table in the projection layer, each word is represented with a d-dimension real-valued vector.
- Calculate the center vector or mean vector of the seed words as followed:

$$\vec{v} = \frac{\sum_{i=1}^k \vec{v}_i}{k} \quad (2)$$

Given k seed words, we denote \vec{v}_i as the word embedding of word i, and \vec{v} as the mean vector of the seed words. Each dimension of \vec{v} is the mathematical expectation of that dimension of all seed words.

- Calculate the variance vector as followed:

$$\vec{D} = (D^1, D^2, \dots, D^i, \dots) \quad (3)$$

$$D^i = \frac{\sum_{j=1}^k (\vec{v}_j^i - \vec{v}^i)^2}{k} \quad (4)$$

We denote \vec{D} as the variance vector, with each dimension D^i as a variance, representing the degree of deviation between the random variable and its mathematical expectation. We also denote \vec{v}^i as the i^{th} dimension of mean vector \vec{v} and \vec{v}_j^i as the i^{th} dimension of vector of word j.

- Sample negative words stochastically in the lookup table, and calculate the negative variance vector \vec{D}_{neg} as step 3.
- Extract feature sub-space with certain dimensions where D^i is small and D_{neg}^i is large. In our model, we gain the error variance vector through computing the difference of \vec{D} and \vec{D}_{neg} and reorder the error variance vector before extracting the first K small dimensions as sub-space representation vector.
- Re-compute the cosine similarity of mean vector and each word using their sub-space representation vector but not the full vector.

4 Experiments and Results

In our method, we actually choose Google's new published tool word2vec to train word embeddings because of its efficiency of learning high-quality distributed representations that capture a large number of syntactic and semantic word relationships. Word2vec's training is extremely efficient [11]: an optimized single machine imple-

mentation can train on more than 100 billion words in one day. We can finish training in several hours of processing 5.6G data after converting the tokens into lower case.

Following most researchers, we choose Wikipedia articles as the corpus to train the model and word embeddings because of their wide range of topics and word usages, and clean organization of document by topics [3]. We use the Wikipedia corpus with a total of 2 million articles and 990 million tokens.

After training the word embeddings, we choose the top 100,000 most frequent words in Wikipedia and implement our experiment based on these words. Each word embedding is a 200-dimension real-valued vector that supposed to represent a word's meaning and features.

In Prior Statistical Knowledge step, we assign 29 POS features to 99203 words, 3 NER features to 82795 words and 94 Parsing features to 64813 words. Some words may not have all three labels because of the statistical computing results.

We give (1) our model's performance in WordNet synonym dictionary in table 3 and (2) our model versus word2vec on the performance of this task in table 4.

Table 3. Our model's performance in WordNet synonym dictionary

Seed words	Our FSS model / word(rank)	WordNet
distressing, sad, pitiful	dreadful(2), miserable(5), painful(6), deplorable	Deplorable, sad, distressing, miserable, pitiful, sorry, painful, dreadful
animal, creature	monster(2), giant(4), cat, alien, ape	animal, creature, monster, giant
acquire, win, gain	obtain(1), make(3) reach(4), achieve(7), retain, get	Acquire, win, gain, attain, obtain, reach, achieve, make, earn

Table 4. Our model versus word2vec on the performance of this task

Seed words	Our model	Word2vec
Beijing, Shanghai, Nanjing	Harbin, Suzhou, Taiwan	Chinese, China, Taiwan
son, woman, lady	widow, lover, man	man, cousin, father
sorry, sad, upset	terrible, ironically, hurt	obviously, feeling, hurt

Table 3 shows that our model can reveal new words having the same meaning as the seed words. The number in the parentheses after each word is the rank of the word among the expand words. Words without a parentheses means they have some common semantic components with the seed words but not appear in the WordNet synonym dictionary.

Table 4 shows that our model can reveal synonymous words of seed words that word2vec cannot and improve the rank of nearest words. More importantly, Word2vec can only find words with high co-occurrence of seed words, but our model can reveal words with the same feature sub-space.

Table 5. Performance of our model versus word2vec on WordNet synonym dictionary

Models	Precision@5	Recall@5	F1@5
Our model	77.20%	80.24%	78.69%
Word2vec	79.45%	73.40%	71.36%
Models	Precision@10	Recall@10	F1@10
Our model	90.40%	92.20%	91.29%
Word2vec	86.30%	87.35%	86.82%

We also give a comparison between the performance of our model and word2vec on WordNet synonym dictionary in table 5.

Our model expands words with several seed words while Word2vec expand words with only one word. To compare the two methods more precisely, we reorder the results of Word2vec on each seed word by averaging each expanded word’s similarity score. In the new rank, words with a better place more likely similar to all seed words. For example, given seed words “Beijing”, “Shanghai”, “Guangzhou”, “capital” is expanded through “Beijing” only, we will divide its weight by 3. While “Chinese” is expanded through all three seed words, we will average the three scores as the new score.

We choose 235 word groups extracted from WordNet synonym dictionary and utilize all words but one of each word group as seed words and the left one as evaluation word. We evaluate two methods on top 5 and top 10 of rank lists. For example, Precision@5 means that evaluation word is in the top 5 of expanded rank list. Table 5 shows that our model outperforms Word2vec on revealing latent common semantic components and expanding synonymy.

5 Conclusion

We introduced a new NLP task similar to word expansion task or word similarity task and proposed a Feature Sub-Space Extraction Model with Prior Statistical Knowledge

and Negative Sampling based on word embeddings. Our problem and methods are significant in information retrieval and natural language processing. It can be used to reveal new similar words, synonymous words and explore the common meaning part of seed words. Our model has been proved to be proper, effective and outperform original word embeddings like word2vec.

References

1. Dagan, I.: Contextual word similarity. *Handbook of Natural Language Processing*, 459-475 (2000)
2. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007* (1995)
3. Huang, E.H., et al.: Improving word representations via global context and multiple word prototypes. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, vol. 1, Association for Computational Linguistics (2012)
4. Budanitsky, A.: *Lexical Semantic Relatedness and Its Application in Natural Language Processing*. Technical Report CSRG-390, Dept. of Computer Science, Univ. of Toronto (August 1999)
5. Lin, D.: An Information-Theoretic Definition of Similarity. In: *Proc. Int'l Conf. Machine Learning* (July 1998)
6. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (2010)
7. Bengio, Y., et al.: Neural probabilistic language models. In: *Innovations in Machine Learning*, pp. 137–186. Springer, Heidelberg (2006)
8. Mikolov, T., et al.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
9. Collobert, R., et al.: Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12, 2493–2537 (2011)
10. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, vol. 13. Association for Computational Linguistics (2000)
11. Gutmann, M.U., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The Journal of Machine Learning Research* 13, 307–361 (2012)
12. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (2013)

Word Vector Modeling for Sentiment Analysis of Product Reviews

Yuan Wang¹, Zhaohui Li^{1,*}, Jie Liu^{1,2},
Zhicheng He¹, Yalou Huang³, and Dong Li⁴

¹ College of Computer and Control Engineering, Nankai University, Tianjin, China

² Information Technology Research Base of Civil Aviation Administration of China,
Civil Aviation University of China, Tianjin, China

³ College of Software, Nankai University, Tianjin, China

⁴ College of Economic and Social Development, Nankai University, Tianjin, China
{yayaniuzi23,295583194,hezicheng}@mail.nankai.edu.cn,
{jliu,huangyl,lidongnk}@nankai.edu.cn

Abstract. Recent years, an amount of product reviews on the internet have become an important source of information for potential customers. These reviews do help to research products or services before making purchase decisions. Thus, sentiment analysis of product reviews has become a hot issue in the field of natural language processing and text mining. Considering good performances of unsupervised neural network language models in a wide range of natural language processing tasks, a semi-supervised deep learning model has been proposed for sentiment analysis. The model introduces supervised sentiment labels into traditional neural network language models. It enhances expression ability of sentiment information as well as semantic information in word vectors. Experiments on NLPCC2014 product review datasets demonstrate that our method outperforms the traditional methods and methods of other teams.

Keywords: Sentiment Analysis, Product Review, Neural Network Language Model, Semi-supervised Learning.

1 Introduction

With the development of the internet environment, more and more people start shopping online, interactions between customers and shopping websites are also more frequent. According to the 33rd China internet network development state report, the user scale of online shopping is 302 million. Customers often write down product reviews on the internet. The reviews contain opinions with different emotional colors and personal semantic information. Thus, mining these product reviews has vital benefit for both consumers and businesses. It helps the consumers to choose the right products and also helps the businesses to improve their products.

* Corresponding author.

However, faced with the huge amounts of product reviews on the internet, it is difficult to grasp the panorama of product reviews accurately and quickly. Thus, mining and analyzing plenty of product reviews becomes a hot issue in recent years. An important research direction is sentiment analysis, which can be classified into sentiment polarity classification[1, 2] and subjectivity classification[3]. Sentiment polarity classification includes binary classification[1](positive and negative) and multivariate classification[2, 4]. This paper mainly researches the binary classification, which makes a judgment of these product reviews by sentiment orientation. There are two main kinds of classification method, the methods based on sentiment knowledge and the methods based on machine learning techniques. The former ones judge texts' sentiment relying on sentiment word dictionary and language rules, while the latter ones adopt machine learning classification techniques with feature selection.

With the prevalence of online language, review texts become more abundant, and emotional expressions also become more unconstrained. Besides the rewrite of homophonic, wrong characters, pictograms in the texts and tones, there are more irony and parody in tone. Due to these informal and diverse texts, traditional methods based on sentiment dictionary now face difficulties, such as frequent update and inaccurate human judgment. Sentiment analysis methods based on feature classification have attracted increasing attention of many researchers in recent years, as they can return feedback quickly and learn from data adaptively. Pang[1] firstly applied machine learning techniques to sentiment analysis, and demonstrated unigram's effectiveness on textual modeling. But unigram models treat words as independent indices in dictionary, which leads any two words have the same semantic relativity. However, it's far from the truth. For example, any two words out of "good", "well" and "China" have the same semantic distance via the unigram model. But the semantic distance between "good" and "well" is closer than that between "good" and "China". This example shows the method's flaws in text semantic expressions. In order to solve the problem, the continuous word vectors can model semantic information effectively[5-7]. Even so, the word vector method still can't distinguish the important sentiment information explicitly. For example, the model can learn out that "good" and "well" have the similar meaning, but can't grasp the information that the both of the two words have strong positive affectivity which is vital for sentiment discrimination. Thus, this work proposes a word vector neural network model, which takes both sentiment and semantic information into account. This word vector expression model not only fuses unsupervised contextual information and sentence level supervised sentiment labels, but also learns words' semantic information and sentiment at the same time. So that it can distinguish sentiments and finish the product reviews sentiment analysis task.

The content of this paper can be summarized as follows: Section 2 introduces the related works of sentiment analysis and the modeling of word vectors. We discuss how to introduce word vectors to help to supervise word modeling

in Section 3. Section 4 introduces the method of model acceleration. Section 5 proposes the word vector method has the ability of sentiment analysis by the experiments. Conclusions are given in Section 6.

2 Related Work

Related researches of this paper mainly include sentiment classification and word semantic modeling.

The mainstream methods of sentiment classification can be divided into methods based on sentiment knowledge and methods based on machine learning techniques. The former ones discriminate sentiment orientation of texts by using universal[8] or domain specific[9] sentiment word dictionaries. Kim et al.[8] used WordNet and HowNet to classify sentiment orientation. Tong[9] manually chose sentiment phrases and developed a domain specific sentiment vocabulary for movie reviews. Besides choosing sentiment words manually, Turney[10] used point-wise mutual information to identify emotional words and their polarities. The later ones use machine learning techniques to learn features of review texts, and then classify sentiment orientation of reviews by using discriminant methods, such as SVM, Naive Bayes, Maximum Entropy and etc. Pang[1] firstly modeled the sentiment polarity discrimination as a binary classification problem, using “n-gram” as features. On this basis, the following researches treat sentiment classification as feature selection tasks. Kim[11] put forward position features in addition to “n-gram” features. What’s more, Pang[4] defined sentiment classification tasks at different fine-grained levels, using multivariate classification and regression classification to complete multi-classification task of sentiment.

For word semantic modeling, recent researches[5–7] show that word representation using continuous vectors can model semantic information effectively, and it has already made great progress[5, 6] in different tasks for natural language processing. The classic methods include the language model of a three-layer neural network based on “n-gram” assumption[5], a language model based on neural networks by scoring a word string of consecutive n words[6], and the recurrent neural network language model[7]. In addition to neural network language models, probabilistic topic models, such as PLSA[12] and LDA[13], learn probability distribution of words under different topics by introducing latent variables, i.e. topics, where the topic distributions of words indicate words’ locations in the topic space.

Taking advantage of continuous word vectors, Maas[14] merged it with supervised emotional information to learn the word vector semi-supervisedly based on probabilistic topic models. Inspired by neural network language models[5–7] and Maas[14], this work considers supervised sentiment information in the learning process of neural network language models, builds a semi-supervised model, and learns the word vectors with both abilities of sentiment orientation expression and semantics expression.

3 Word Vector Model for Sentiment Analysis

3.1 Neural Network Language Model

In neural network language model, every word w represents a vector $C(w)$, and the current word is predicted by words around it. The neural network model here is inspired by three layered neural network language model (NNLM) proposed by Bengio[5] in 2003. NNLM follows the n-gram assumptions, where current word is only related to $n - 1$ words before it. So the generation probability of word is $P(w_t = i) = P(w_t = i|w_{t-n+1}^{t-1})$. And the generation probability of sentence with N_d words is:

$$P(w_1^{N_d}) = \prod_{t=1}^{N_d} P(w_t|w_1^{t-1}) \propto \prod_{t=1}^{N_d} P(w_t|w_{t-n+1}^{t-1}), \tag{1}$$

where $w_i^j = (w_i, w_{i+1}, \dots, w_{j-1}, w_j)$ is a word sequence from w_i to w_j . The likelihood optimization of the model is shown in Formula 2.

$$\max \sum_d \sum_{t=1}^{N_d} \log P(w_t|w_{t-n+1}^{t-1}) \tag{2}$$

Illustrated in figure 1, the model has three layers. At the bottom is the input layer, including $n - 1$ word vectors. The hidden layer in the middle contains hidden nodes with a linear projection input and a nonlinear projection output. The upper output layer is a predict layer with the same size as the dictionary.

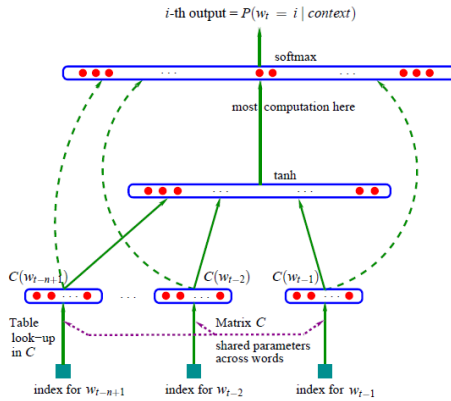


Fig. 1. Neural Network Language Model[5]

NNLM predicts word w_t in two main steps:

- 1 Words are mapped to m dimension vectors $C(w)$, C is a $|V| \times m$ word vector matrix. $|V|$ is the size of dictionary. The transform from word w to $C(w)$ is to extract the line indexed by w from the C .

- 2 The $n - 1$ word vectors before w_t are connected to form the $(n - 1) \times m$ dimension input, denoted as x . The output is a $|V|$ -dimension vector, where element i is estimation probability of $P(w_t = i)$, denoted as y . The whole process can be expressed as:

$$f(i, w_{t-n+1}, \dots, w_{t-1}) = g(i, C(w_{t-n+1}), \dots, C(w_{t-1})) \quad (3)$$

For the hidden layer, the input is $d + Hx$ instead of x in traditional neural network, and the output is a $|V|$ dimension vector y activated by \tanh function, y_i is the value of the node i . The input of the output layer is the log prediction probability of the next word, the output is the normalization probability through a *softmax* activation function. The calculation of y is:

$$y = b + Wx + U * \tanh(d + Hx), \quad (4)$$

where $x = (C(w_{t-n+1}), \dots, C(w_{t-1}))$. The probability of the next word is:

$$P(w_t | w_{t-n+1}^{t-1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}}. \quad (5)$$

We estimate parameters with the gradient descent method. C is randomly initialized, (b, W, U, d, H) are all initialized with 0.

This paper adopts the NNLM structure in semantic learning, but semantic modeling of word vectors uses Skip-gram model[15]. The main differences between are, 1) Skip-gram model uses current word to predict other words in a certain range window around it, 2) mapping from the input layer to the hidden layer is simplified, where vector of w_t is the output of the hidden layers, i.e. $x_t = C(w_t)$. Given the current word w_t and the window size c , the maximization goal of the Skip-gram model is $\sum_d \sum_{t=1}^{N_d} \sum_{-c \leq j \leq c, j \neq 0} P(w_{t+j} | w_t)$. When window size is 5, the model makes likelihood probability largest when predicting the ten words around the current word, and smallest for other words. After simplification, the transformation from input to output turns into $y = b + Wx$.

3.2 Learning Word Sentiment

NNLM can grasp the semantic information of words, but can't consider supervised sentiment information for sentiment analysis. Thus, we use sentiment orientation information of reviews to help in the learning of words' sentiment polarity.

Since the prediction relationship between words is linear, we adopt a logistic regression linear model to model sentiment orientation of each word. First, we map sentiment labels into $[0, 1]$, denoted as s . For binary sentiment discrimination, $s = 1$ means positive and $s = 0$ means negative. Assume that every word has the same capability to express sentiment independently, the generation probability of a sentence sentiment label is:

$$P(s|d; C, a, b_s) = \frac{1}{N_d} \sum_{t=1}^{N_d} P(s|w_t; C, a, b_s), \quad (6)$$

where a is a m -dimension logistic regression parameter vector, b_s is bias of sentiment orientation. This logistic regression model defines sentiment orientation of the words as a probabilistic classification problem of a hyperplane:

$$P(s|w_t; C, a, b_s) = \sigma(aC(w_t) + b_s), \quad (7)$$

where the mapping method of w_t is the same as the Skip-gram model. $\sigma(x)$ is sigmoid activation function. The objective function is:

$$\max_d \sum_d \frac{1}{N_d} \sum_{t=1}^{N_d} P(s|w_t; C, a, b_s). \quad (8)$$

This model makes words with similar sentiment orientation closer in the semantic space, and can be separated by a hyperplane.

3.3 Word Vector Model with Emotional Supervision

We combine the original semantic learning part (in Section 3.1) and the supervised sentiment learning part (in Section 3.2) to learn semantic information and sentiment polarity characteristics of words simultaneously. And the combined objective function is:

$$L = \sum_d \sum_{t=1}^{N_d} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j}|w_t; C, W, b) + \beta \sum_d \frac{1}{N_d} \sum_{t=1}^{N_d} \log P(s|w_t; C, a, b_s), \quad (9)$$

where β is a parameter to balance two parts. The left part comes from the Skip-gram model used for semantic unsupervised learning, and the right part comes from the logistic regression linear model for sentiment supervised learning. The two parts share the word vector C , so that word vector learning are guided by the sentence sentiment label s . The parameters need to learn is:

$$\theta = (C, W, b, a, b_s). \quad (10)$$

The parameters are $|V| \times m$ word vector matrix C showing semantic information with sentiment orientation, neural network's parameters W (a $m \times |V|$ matrix) and output biases b (a $|V|$ -dimension vector), regression parameters a (a m -dimension vector) and bias b_s (a value) in the sentiment supervised learning part.

We learn parameters with stochastic gradient ascent methods. The update law is:

$$\theta \leftarrow \theta + \varepsilon \frac{\partial L}{\partial \theta}, \quad (11)$$

where ε is a learning rate. Moreover, multi-thread technology can be used to share model parameters and finish learning rapidly, because every update is independent for different reviews.

4 The Optimization of the Model

As we can see from Figure 1, we need to do an iterative calculation of all output nodes because of the softmax method used by output layer in the prediction of NNLM, which lowers the performance of model a lot. At the same time, the complexity of our model increases exponentially with $|V|$ increases. To reduce the computational complexity of output layer prediction, we use a hierarchical softmax method.

Hierarchical softmax, using Huffman coding to code output layer as a node tree, makes coding of high frequency words shorter and low frequency ones longer. The maximum depth of the node tree is $\log|V|$. When predicting output and back-propagation gradient, we just need to calculate node error on the route. Prediction to every word just calculates values of nodes and residuals at most, which can improve the performance a lot. Details are in [16]. In addition, the tree encoding can adopt any hierarchical agglomerative clustering method to lessen the number of output nodes and improve performance.

This strategy mainly ignores normalizing term in softmax without normalized operation. It sees the output of the last layer as a classification task, which aims to make the model classify correctly and automatically. Concretely, it classifies the input x correctly to the category containing y . Hierarchical softmax expresses the category containing y paired by x as a tree structure, the training processing is doing classification layer by layer until reach the leaf nodes and simplifies the number of nodes needed to predict in the output layer. Benefiting from hierarchical softmax, the performance of the model promotes a lot.

5 Experiments

5.1 Dataset

This paper adopts the product review dataset in NLPCC2014 “Sentiment Classification with Deep Learning Technology” task to validate our model. The dataset is divided into training and testing sets. The training dataset contains 20,000 product reviews from product review websites, including 10,000 Chinese reviews and 10,000 English reviews. The datasets in different language are balanced, and each contains 5,000 positive samples and 5,000 negative samples. The testing dataset includes 5,000 reviews, including 2,500 product reviews for each language.

As our model analyzes sentiment orientation of product reviews by word vectors, we takes some necessary preprocessing for these informally written product reviews. 1) Remove all hyperlinks, such as <http://music.jschina.com.cn/adsu.asp?id=385&userid=62039>. 2) Remove the special characters in html, such as `<`, `>` and etc. 3) Use the Yebol Chinese word segmentation platform¹ to segment the sentences in Chinese. After that, words and punctuation

¹ <http://ics.swjtu.edu.cn/>

characters with definite emotion, like exclamatory marks, question marks, apostrophes and tildes, in reviews are left.

The format of the raw data is XML, taking an English negative sample as an example: `<review id="9214">The Rice cooker works great. Missing the Steaming Basket. Please send it asap. That's the reason I purchased because it said "Rice Cooker/Steamer" Been waiting too long</review>`. This example reflects the complexity of sentiment analysis, as well as the shortages of traditional method based on sentiment words and grammatical rules. If we take the sentiment dictionary methods, "great" is an obvious positive word, and is not modified by negative words or adversative conjunctions. So this sentence is likely to be judged as positive, which leads to a mistake.

5.2 Quantitative Evaluation

In training phase of the sentiment classification task, we use classification accuracy (*Acc*) as our metric for guiding parameter selection quickly. In testing phase, metrics are based on precision (*P*), recall (*R*) and F1. Let TP be the number of the positive samples which are correctly classified, FP be the number of the negative samples which are falsely classified, TN be the number of the negative samples which are correctly classified, FN be the number of the positive samples which are falsely classified. The evaluation metrics are as follows.

$$\begin{aligned}
 P_{pos} &= \frac{TP}{TP + FP}, R_{pos} = \frac{TP}{TP + FN}, F1_{pos} = \frac{2P_{pos}R_{pos}}{P_{pos} + R_{pos}} \\
 P_{neg} &= \frac{TN}{TN + FN}, R_{neg} = \frac{TN}{TN + FP}, F1_{neg} = \frac{2P_{neg}R_{neg}}{P_{neg} + R_{neg}} \\
 Acc &= \frac{TP + TN}{TP + FP + TN + FN}
 \end{aligned} \tag{12}$$

5.3 Experiment Results

To verify our model's effectiveness in sentiment analysis, we compare different features as text representation, and then use a linear support vector machine² with 5-fold cross validation to classify sentiment orientation of product reviews. After learning word vectors, we take the same way as [14] to represent reviews. In training phase, 1) we compare our features with term frequency features [1] (denoted as *TF*) and sentiment knowledge features (denoted as *Senti*), 2) we combine the best two sets of features obtained in experiment 1) and do parameter selection. In testing phase, the parameters that achieve the best performance in training phase are picked out to predict sentiment orientation of product reviews in the test data set.

Among those features, *TF* use term frequency with the *tfidf* schema as reviews' features and *Senti* is composed of 17 features based on sentiment knowledge. They are 1-2) the number of positive/negative words, 3-9) the number

² <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

of verbs/nouns/adjectives/exclamations/ellipses/tildes/question marks, 10-11) the number of positive/negative phrases, 12) emotional index of the whole review, 13-14) emotion index of the first/last sentence, and 15-17) the number of positive/negative/neutral sentences. Here we use HowNet as the sentiment dictionary. The 10-12 dimensions are obtained by a modified factor method[17], and the 13-17 dimensions are obtained by a rule method[18]. For *Senti*, we find classification accuracy all falls about 2%-6% after removing some of the dimensions. Due to space limitations, these experiments are not listed specifically. So, we use all 17-dimensional features as *Senti* in the experiments below.

Experimental Results in Training Phase. Table 1 shows the results on only one kind of features, which are word frequency features, sentiment knowledge features and features we proposed. From Table 1, we have following conclusions. (1) Each model’s classification accuracy in English reviews are higher than those in Chinese ones by about 9%, that shows the complexity of Chinese and difficulties in sentiment analysis of Chinese reviews. (2) For English reviews, *TF* has achieved the best results, with accuracy at 81.48%, 2.26% higher than our model. (3) For Chinese reviews, our model has achieved the best, with accuracy at 71.45%, 1.39% higher than *TF*. Thus, our model is more adaptable to the complex Chinese texts than other models. The reason is that our model is powerful to learn sentiment orientation of words as well as semantic expression automatically, especially when users express their emotion with informal and short language fragments showing in Chinese reviews. However, English reviews three times longer than Chinese reviews on average, in which users emphasis their emotions by repeating some key words. So *TF* acts enough good. Meanwhile, the results based on the sentiment knowledge features are the worst. The main reason is that online language is not standardized, full of rhetoric, strongly arbitrary and changes quickly, which also makes approaches based on manually constructed sentiment knowledge no longer applicable and unable to meet the sentiment characteristics of review texts.

Table 1. Classification accuracy with different features

Dataset	Senti	TF	Our Model
English	75.68%	81.48%	79.22%
Chinese	66.37%	70.06%	71.45%

In our model, there are two important parameters, the dimension m of word vector $C(w)$ and the balance factor β between the language model part and sentiment discrimination part. Figure 2 shows the classification accuracy when parameters change.

Figure 2 shows that classification accuracy on the English dataset grows with the increase of the vector dimension, and reaches a maximum 0.7922 when $m = 900$ and $\beta = 0.01$. Performances are similar when β equals to 0.01 and 0.1. However, when β equals to 0.01, accuracy is 0.04% higher by average.

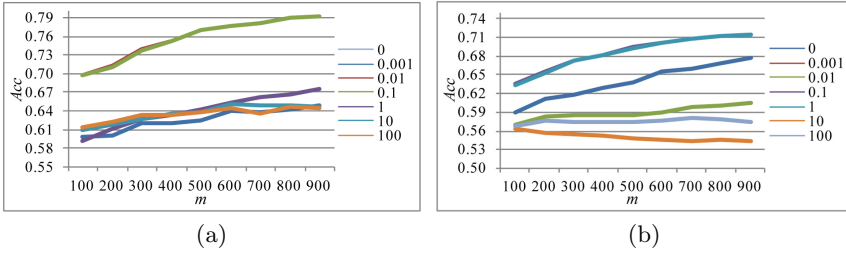


Fig. 2. Classification accuracy with different parameters on (a) English data and (b) Chinese data

In the Chinese dataset, the highest accuracy is 0.7145 when $m = 900$ and $\beta = 1$. Performances are similar when the balance factor β equals to 0.1 and 1. However when β equals to 0.1, classification accuracy is 0.05% higher by average. Compared to performance on the English dataset, (1) classification accuracy decreases with the increase of the vector dimension except when β equals to 10, (2) the performance of the model fluctuates with the increase of the vector dimension when β equals to 100. The other results are all similar to that of the English dataset.

The experiments above tell that sentiment knowledge features performed poorly, while word frequency features and our model show their own strengths on data sets in different language. So we consider a combination of these two features to accomplish the task. After parameter selection, the accuracy does have a great improvement. Among them, classification accuracy on English data reaches 85.16% when $m = 200$ and $\beta = 0.1$, 4.28% higher than the single feature method; that on Chinese data reaches 77.40% when $m = 500$ and $\beta = 0.1$, 5.5% higher than the single feature method. See Figure 3 for details.

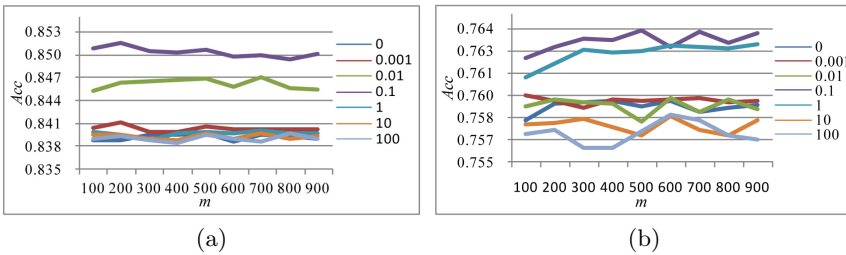


Fig. 3. Classification accuracy by using combination features on (a) English data and (b) Chinese data

The results show that TF increases the dimension of reviews' features, which makes accuracy less sensitive to vector dimension. When β is fixed, accuracy fluctuates, especially in the Chinese data set. In both data sets, accuracy performs the best when $\beta = 0.1$. When β equals to 0, the model degenerates into the original Skip-gram model, and the emotional discrimination performs worst.

It verifies the effectiveness of the semi-supervised word vector learning model with emotional discriminant information proposed in this paper.

Experimental Results in Testing Phase. Known from the experimental results in training phase, the classification accuracy is higher when using both word frequency features and features proposed in this paper. Thus, we combine the two kinds of features, and select two sets of parameters that achieve the best results on training dataset to predict labels for the testing datasets. Table 2 gives the evaluation results on testing datasets.

Table 2. Classification results on test data and parameter selection

Dataset	Parameters	Negative			Positive		
		P	R	F1	P	R	F1
English	$m = 200, \beta = 0.1$	0.864	0.855	0.860	0.856	0.866	0.861
	$m = 100, \beta = 0.1$	0.865	0.851	0.858	0.853	0.867	0.860
Chinese	$m = 500, \beta = 0.1$	0.780	0.748	0.764	0.758	0.789	0.773
	$m = 700, \beta = 0.1$	0.678	0.452	0.545	0.591	0.794	0.678

We know from the table that both results of the English dataset in the testing phase are slightly better than those in the training phase, while both results of the Chinese dataset in the testing phrase are worse than those in the training phase. However, the overall levels are almost the same, which verifies the effectiveness of our model.

6 Conclusion

This paper proposes a word vector learning method aiming at sentiment analysis and explores the performance of deep learning models on the task of sentiment classification for product reviews. The model solves the problem that the traditional unsupervised word model can't catch sentiment orientation information of words. The method introduces label information into the unsupervised neural network language model, so that the sentiment orientation and topical semantic information are both learned. We also accelerate our neural network language model. The experiments on the NLPC2014 emotional evaluation dataset show the effectiveness of our model.

In the future, we intend to explore this model's performance under lots of unsupervised data when a little supervised data exists. Besides, our model is a general semi-supervised learning framework modeling sentiment label information as well as unsupervised semantic information. We want to explore the learning ability of the model on different semantic aspects of words when treating different label information as different guide information.

Acknowledgments. This research is supported by the National Natural Science Foundation of China (No. 61105049 and No. 61300166), the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201303), the Natural Science Foundation of Tianjin (No. 14JCQNJC00600) and the Science and Technology Planning Project of Tianjin (No. 13ZCZDZX01098).

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, vol. 10, pp. 79–86. Association for Computational Linguistics, Stroudsburg (2002)
2. Linhong, X., Hongfei, L., Jing, Z.: Construction and analysis of emotional corpus. *Journal of Chinese Information Processing* 22, 116–122 (2008)
3. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL 2004, pp. 271–278. Association for Computational Linguistics, Stroudsburg (2004)
4. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, pp. 115–124. Association for Computational Linguistics, Stroudsburg (2005)
5. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. *Journal of Machine Learning Research* 3, 1137–1155 (2003)
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 160–167. ACM, New York (2008)
7. Tomáš, M., Martin, K., Lukáš, B., Jan, C., Sanjeev, K.: Recurrent neural network based language model. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, Makuhari, Chiba, Japan, pp. 1045–1048 (2010)
8. Kim, S.M., Hovy, E.: Automatic detection of opinion bearing words and sentences. In: Companion Volume to the Proceedings of IJCNLP-05, the Second International Joint Conference on Natural Language Processing, Jeju Island, KR, pp. 61–66 (2005)
9. Tong, R.M.: An operational system for detecting and tracking opinions in on-line discussion (September 2001)
10. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, pp. 417–424. Association for Computational Linguistics, Stroudsburg (2002)
11. Kim, S.M., Hovy, E.: Automatic identification of pro and con reasons in online reviews. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL 2006, pp. 483–490. Association for Computational Linguistics, Stroudsburg (2006)
12. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999, pp. 50–57. ACM, New York (1999)

13. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)
14. 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 - Volume 1, HLT 2011*, pp. 142–150. Association for Computational Linguistics, Stroudsburg (2011)
15. Mikolov, T., Tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*, pp. 746–751. Association for Computational Linguistics (May 2013)
16. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 246–252. Citeseer (2005)
17. Jिंगgang, W., Xiao, Y., Dandan, S., Lejian, L., Wei, Z.: A chinese bag-of-opinions method for weibo sentiment analysis. In: *Proceedings of the First CCF Conference on Natural Language Processing and Chinese Computing*, pp. 1–6 (2012)
18. Lixing, X., Ming, Z., Maosong, S.: Hierarchical structure based hybrid approach to sentiment analysis of chinese micro blog and its feature extraction. *Journal of Chinese Information Processing* 26, 73–83 (2012)

Cross-Lingual Sentiment Classification Based on Denoising Autoencoder

Huiwei Zhou, Long Chen, and Degen Huang

School of Computer Science and Technology,
Dalian University of Technology, Dalian 116024, Liaoning, China
{zhouhuiwei, huangdg}@dlut.edu.cn, chenlong.415@mail.dlut.edu.cn

Abstract. Sentiment classification system relies on high-quality emotional resources. However, these resources are imbalanced in different languages. The way of how to leverage rich labeled data of one language (source language) for the sentiment classification of resource-poor language (target language), namely cross-lingual sentiment classification (CLSC), becomes a focus topic. This paper utilizes rich English resources for Chinese sentiment classification. To eliminate the language gap between English and Chinese, this paper proposes a combination CLSC approach based on denoising autoencoder. First, two classifiers based on denoising autoencoder are learned respectively in English and Chinese views by using English corpus and English-to-Chinese corpus. Second, we classify Chinese test data and Chinese-to-English test data with the two classifiers trained in the two views. Last, the final sentiment classification results are obtained by the combination of the two results in two views. Experiments are carried out on NLP&CC 2013 CLSC dataset including book, DVD and music categories. The results show that our approach achieves the accuracy of 80.02%, which outperforms the current state-of-the-art systems.

Keywords: cross-lingual sentiment classification, combination, denoising autoencoder.

1 Introduction

Sentiment classification technique is the task of predicting sentiment polarity for a given text. It could help to mine public opinion from product reviews. Along with the rapid expansion of user-generated information, sentiment classification plays a key role in analyzing a large number of sentiment reviews on the Web.

Researches on sentiment classification have been developed rapidly. Generally, sentiment classification approaches can be divided into two categories: lexicon based approach and machine learning based approach. Lexicon based approach extracts the sentiment words in texts, and identifies the positive or negative polarities of texts based on the sentiment words' polarities in lexicon [1-2]. This method is easily implemented, and could achieve a reasonable accuracy on the foundation of an elaborate lexicon. However, it is difficult to identify the polarities of sentiment words in texts since their polarities may be changed in different context [3]. Machine learning based

approach trains sentiment classifiers based on the context of sentiment words [4-6]. This approach identifies the polarities of texts more accurately than lexicon based approach and is widely used in the sentiment classification task now. However, the method heavily relies on the quality and quantity of corpora, which are considered as valuable resources in sentiment classification task.

Sentiment classification in English has been studied for a long time, and many labeled data for English sentiment classification are available on the Web. However, labeled data in different languages are very imbalanced. The lack of sentiment resources limits the research progress in some languages. In order to overcome this obstacle, cross-lingual sentiment classification (CLSC) [7-10] is proposed, which leverages resources on one language (source language) to resource-poor language (target language) for improving the performance of sentiment classification on target language.

Machine translation services are usually adopted to eliminate the gap between source language and target language in CLSC task. Wan [7] proposed a co-training approach for CLSC, which leveraged an English corpus for Chinese sentiment classification in two independent views: English view and Chinese view. To further improve the performance of co-training approach, Gui et al. [8] incorporated bilingual cross-lingual self-training and co-training approaches by estimating the confidence of each monolingual system. To reduce the impact of translation errors, Li et al. [9] selected high-quality translated samples in the source language.

These methods all adopted shallow learning algorithms, which were optimized based on limited computing units. However, shallow learning algorithms can hardly learn the kind of complicated functions that can discover intermediate representation of the input [11]. Deep learning algorithms could learn intermediate representations through multi-layer non-linear operations in the function learning [12-14]. Searching the parameter space of deep architecture is a difficult task. Bengio et al. [11] proposed an optimization principle to solve this problem, which has worked well for deep belief network (DBN) and autoencoder (AE) [15]. To further make the learned representations robust to partial destroyed input, Vincent et al. [16] raised denoising autoencoder model.

Deep learning has gained huge success in many real-world applications such as computer vision [17] and speech recognition [18]. Collobert et al. [19] applied deep learning to natural language processing (NLP), and proposed a multi-task learning system SENNA including part-of-speech (POS) tagging, chunking, named entity recognition, and semantic role labeling. Tang et al. [20] and Zhou et al. [21] studied the use of deep learning for representation learning on sentiment classification. The representation features learned from deep learning could improve the performance of sentiment classification effectively.

In CLSC task, the language gap between the original language and the translated language greatly influences the classification performance. This paper proposes a combination CLSC approach based on denoising autoencoder to decrease the effects of noisy translated examples from two aspects. On the one hand, denoising autoencoder is adopted to improve the robustness to translation noises. On the other hand, two classifiers are trained in English view and Chinese view, respectively. The final

results are obtained by combining the two classification outputs to eliminate the language gap. Though this paper leverages English corpora for Chinese sentiment classification, the proposed CLSC approach can be applied to other languages. In our systems, χ^2 (CHI) statistical method is used to select sentiment word features, and TF-IDF method is used to set feature weights. The experiments are conducted using NLP&CC 2013 CLSC dataset. The experimental results show that the proposed approach outperforms the current state-of-the-art systems.

The rest of this paper is organized as follows: Section 2 describes the structure of denoising autoencoder. Section 3 presents the combination CLSC approach in detail. Section 4 reports experimental results and analysis. Section 5 concludes our work and gives the future work.

2 Denoising Autoencoder

The traditional autoencoder maps an input vector $\mathbf{x} \in [0,1]^d$ to a hidden representation $\mathbf{y} \in [0,1]^{d'}$, through a deterministic mapping $\mathbf{y} = f_{\theta}(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b})$, parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$. \mathbf{W} is a $d' \times d$ weight matrix, \mathbf{b} is a bias vector. Then, \mathbf{y} is mapped back to a reconstructed vector $\mathbf{z} \in [0,1]^d$ in input space $\mathbf{z} = g_{\theta'}(\mathbf{y}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$ with parameters $\theta' = \{\mathbf{W}', \mathbf{b}'\}$, where $\mathbf{W}' = \mathbf{W}^T$. In the training phase, each training sample $\mathbf{x}^{(i)}$ is mapped to a latent representation $\mathbf{y}^{(i)}$ and a reconstruction $\mathbf{z}^{(i)}$. The parameters of this model are optimized by stochastic gradient descent (SGD) algorithm to minimize the average reconstruction error:

$$\theta^*, \theta'^* = \arg \min_{\theta, \theta'} \frac{1}{N} \sum_{i=1}^n L(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) = \arg \min_{\theta, \theta'} \frac{1}{N} \sum_{i=1}^n L(\mathbf{x}^{(i)}, g_{\theta'}(f_{\theta}(\mathbf{x}^{(i)}))) \quad (1)$$

where N is the number of training samples, L is a loss function and usually defined as reconstruction cross-entropy [18]:

$$L_H(\mathbf{x}, \mathbf{z}) = H(B_{\mathbf{x}} \parallel B_{\mathbf{z}}) = - \sum_{k=1}^d [\mathbf{x}_k \log \mathbf{z}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{z}_k)] \quad (2)$$

Denoising autoencoder is proposed as a modification of the autoencoder and enforces robustness to partially destroyed inputs. The training process of denoising autoencoder is shown in Fig. 1.

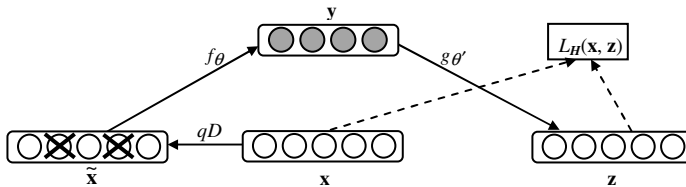


Fig. 1. Structure of denoising autoencoder

For each initial input vector \mathbf{x} , a partially destroyed version $\tilde{\mathbf{x}}$ is acquired by means of a stochastic mapping $\tilde{\mathbf{x}} \sim qD(\tilde{\mathbf{x}} | \mathbf{x})$. A fixed number νl of input components are randomly chosen, and their values are forced to 0, while the others are not changed. ν is the destruction fraction, by which the desired noise level could be adjusted. Denoising autoencoder maps $\tilde{\mathbf{x}}$ instead of \mathbf{x} to a hidden representation \mathbf{y} through a deterministic mapping $\mathbf{y} = f_{\theta}(\tilde{\mathbf{x}}) = s(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b})$, from which we reconstruct a $\mathbf{z} = g_{\theta'}(\mathbf{y}) = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$. The loss function L_H is also defined as formula (2), where \mathbf{z} is a deterministic function of $\tilde{\mathbf{x}}$ rather than \mathbf{x} .

3 The Combination CLSC Approach Based on Denoising Autoencoder

3.1 The Combination CLSC Algorithm

The combination CLSC approach trains sentiment classifier based on English view and Chinese view, respectively. This paper utilizes the labeled English reviews to classify unlabeled Chinese reviews. The framework of the proposed approach consists of a training phase and a classification phase as shown in Fig. 2. In training phase, English labeled reviews are used to train an English classifier in English view. Meanwhile, English-to-Chinese translated labeled reviews are used to train a Chinese

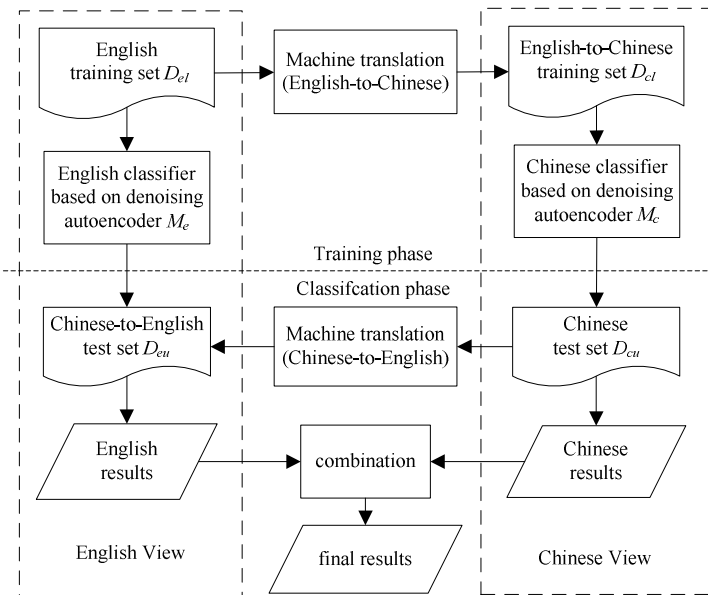


Fig. 2. Framework of the combination CLSC approach

classifier in Chinese view. In classification phase, the English classifier is applied to classify Chinese-to-English translated reviews in English view. Similarly, the Chinese classifier is applied to classify Chinese reviews in Chinese view. The two classification results from English and Chinese views are combined into the final results by comparing positive and negative possibilities for each review.

The combination CLSC algorithm is described in details in the following:

Algorithm. Combination CLSC Algorithm

Input: English training set D_{el} , and its English-to-Chinese version D_{cl} ;

Chinese test set D_{cu} , and its Chinese-to-English version D_{eu} .

For each review R_i :

1. Train English classifier M_e based on D_{el} ;
 2. Use M_e to classify R_i in D_{eu} ;
 3. Obtain the positive possibility $P_e(R_i)$ and the negative possibility $N_e(R_i)$;
 4. Train Chinese classifier M_c based on D_{cl} ;
 5. Use M_c to classify R_i in D_{cu} ;
 6. Obtain the positive possibility $P_c(R_i)$ and the negative possibility $N_c(R_i)$;
 7. Calculate the positive possibility: $P(R_i) = (P_e(R_i) + P_c(R_i)) / 2$;
 8. Calculate the negative possibility: $N(R_i) = (N_e(R_i) + N_c(R_i)) / 2$;
 9. If $(P(R_i) > N(R_i))$, then output positive review R_i ;
 10. Else output negative review R_i ;
-

3.2 Feature Setting

(1) Sentiment Word Features Selection

Words in sentiment lexicon are too many to be all used as sentiment word features. Statistical methods are usually adopted to select effective sentiment word features. This paper investigates the influence of feature selection based on high-frequency words method and CHI statistical method respectively.

High-Frequency Words Method: This method selects the top 2000 high-frequency words as sentiment word features.

CHI Statistical Method: This method enables to measure the association between feature t_i and class C_j [22]. Chi-square value between feature t_i and class C_j is defined as follows:

$$\chi^2(t_i, C_j) = \frac{N \times (A \times D - B \times C)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \quad (3)$$

where A , B , C , and D denote the co-occurrence frequency between a feature t_i and a class C_j . They are corresponding to the case of (t_i, C_j) , (t_i, \bar{C}_j) , (\bar{t}_i, C_j) , (\bar{t}_i, \bar{C}_j) , while N is the sum of them. If $A \times D - B \times C > 0$, feature t_i has a positive correlation

with C_j , otherwise a negative correlation. Therefore, the features with higher CHI-square values $\chi^2(t_i, C_j)$ will be more discriminative. This paper selects the top 2000 words with high CHI-square values as sentiment word features.

(2) Negation Features

Some sentiment words are often modified by negation words, which leads to inversion of their polarities. We take into account 14 usually used negation words in English such as “not”, “without”, and “none”; 5 negation words in Chinese such as “不” (no/not), “不会” (cannot), and “没有” (without). A sentiment word modified by these negation words in the window $[-2, 2]$ expresses the opposite polarity. A negation feature is introduced to each sentiment word to represent the negative form of this word. We simply insert negation feature in front of each sentiment word in a feature vector. Meanwhile, sentiment word features remain the initial meaning. If there is no negation word in the window, the value of negation feature is set to 0. The feature vector containing negation features is extended to 4000 dimensions:

$$vector = (neg_1, sent_1, \dots, neg_i, sent_i, \dots, neg_{2000}, sent_{2000}), (i=1, 2, \dots, 2000) \quad (4)$$

where $sent_i$ denotes sentiment word feature, and neg_i represents negation feature.

(3) The Feature Weight Calculation

The following three methods of feature weight calculation are investigated in this paper: Boolean method, Term-frequency method, and TF-IDF method.

Boolean Method: This method thinks that all feature words have the same importance to classification, and sets a feature weight to 0 or 1 only depending on whether it appears in a review.

Word Frequency Method: This method takes word frequency into account in sentiment classification. In our work, the frequency of each word in each review is calculated and normalized as feature weight.

TF-IDF Method: This method takes word frequency (TF) and inverse document frequency (IDF) into consideration [23]. The feature weight is calculated by the following formula:

$$w_{ij} = tf_{ij} \times \log \frac{N}{n_i} \quad (5)$$

where N is the total number of reviews in corpus, tf_{ij} is the frequency of the word i occurring in the review j , n_i is the number of reviews containing word i . In this way, word frequency and review frequency of word are both considered in weight calculation.

Besides, the sentiment words in the summary are more important than other parts of a review. To highlight the importance of these words, sentiment words surrounded by the “<summary>” tag in each review are counted twice in the frequency calculation in word frequency method and TF-IDF method.

4 Experimental Results and Analysis

4.1 Experimental Settings

The proposed approach is evaluated on NLP&CC 2013 CLSC dataset^{1,2}. The dataset consists of reviews on three categories: Book, DVD and Music. Each category contains 4,000 English labeled data (ratio of positive and negative examples is 1:1), 4,000 Chinese unlabeled data.

In the experiments, *Google Translate*³ is adopted for both English-to-Chinese translation and Chinese-to-English translation. Geniatagger⁴ is used as POS tagging tool and ICTCLAS⁵ is used as Chinese word segmentation tool. Denoising autoencoder is developed based on Theano system⁶ [24].

The architecture of denoising autoencoder used in our experiments is 4000-500-2, which represents the number of units in input layer is 4000, in a hidden layer is 500, and in output layer is 2. The output layer is a softmax layer, where 2 units denote the possibility scores of a review being positive and negative, and their sum is 1. For layer-wise unsupervised learning, we train the weights of each layer independently with the fixed number of epochs equal to 30 and the learning rate is set to 0.1.

The performance is evaluated by the correct classification accuracy for each category, and the average accuracy of three categories, respectively. The category accuracy is defined as:

$$Accuracy_c = \frac{\# system_correct}{4000} \quad (6)$$

where c is one of the three categories, and $\# system_correct$ stands for the number of correctly classified reviews in c .

The average accuracy is defined as:

$$Accuracy = \frac{1}{3} \sum_{i=1}^3 Accuracy_c \quad (7)$$

4.2 Evaluation on Combination CLSC Approach

In this section, the experiments evaluate the performance of our feature selection methods and combination CLSC approach. The destruction fraction ν is set to 0.1.

¹ <http://tcci.ccf.org.cn/conference/2013/dldoc/evsam03.zip>

² <http://tcci.ccf.org.cn/conference/2013/dldoc/evdata03.zip>

³ <http://translate.google.cn/>

⁴ <http://www.nactem.ac.uk/tsujii/GENIA/tagger/>

⁵ http://www.ictclas.org/ictclas_download.aspx

⁶ <http://deeplearning.net/software/theano/>

(1) Effect of Sentiment Word Features Selection

We first evaluate the classification performance in English and Chinese systems, respectively. Table 1 shows effect of sentiment word features selection methods in English and Chinese systems. As shown in Table 1, CHI statistical method outperforms high-frequency words method in DVD and Music categories. The average accuracies of CHI statistical method are 0.46% and 0.50% higher than high-frequency words method in English and Chinese systems, respectively. From these results, we can conclude that CHI statistical method is more effective than high-frequency words method in classification task. The sentiment word features selected with CHI statistical method are used in the following experiments.

Table 1. Effect of Sentiment Word Features Selection

System	Methods	Book	DVD	Music	Accuracy
English	High-frequency	74.53%	75.43%	73.8%	74.58%
	CHI statistic	73.03%	76.93%	75.15%	75.04% (+0.46%)
Chinese	High-frequency	78.40%	74.45%	73.15%	75.33%
	CHI statistic	78.15%	75.05%	74.30%	75.83% (+0.50%)

(2) Effect of Negation Features

The performance of classification systems with or without negation features in Chinese and English systems are given in Fig. 3, respectively. The performance with negation features are steadily better than the performance without them. In most cases, the improvement is considerable. Negation features improve the accuracies by 2.37% (from 73.46% to 75.83%) in Chinese systems and 4.79% (from 70.25% to 75.04%) in English systems. Negation features in English systems are more effective than those in Chinese systems, which perhaps because the English negation words used in English systems are more than in Chinese systems. Negation features are employed in our following experiments.

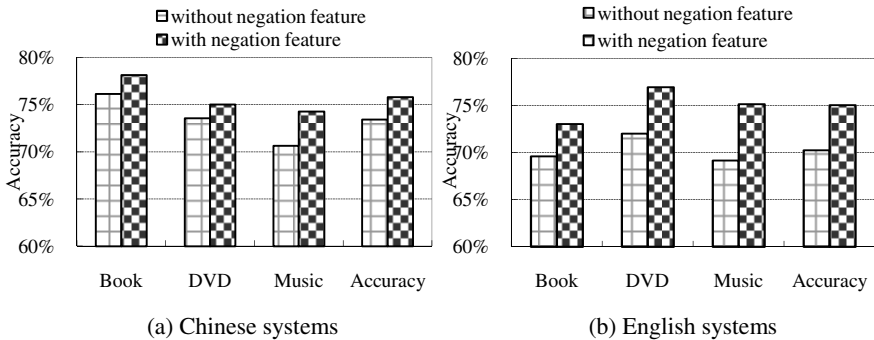


Fig. 3. Performance comparison with or without negation features

(3) Effect of Feature Weight Calculation Methods

Boolean, word frequency and TF-IDF feature weight calculation methods are compared in Fig. 4. Generally, TF-IDF method achieves the best accuracy, while Boolean method performs poorly in sentiment classification task. TF-IDF method would reflect the latent contribution of feature words to the reviews, compared with Boolean and word frequency methods. The feature setting of CHI feature selection method together with TF-IDF weight calculation method achieves the best performance in all categories, which are exploited for the following combination CLSC systems.

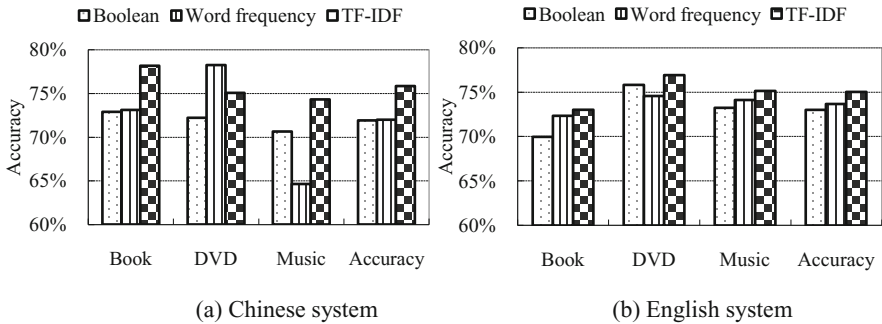


Fig. 4. Performance comparison with different weight calculation methods

(4) Performance of Combination CLSC Systems

Table 2 evaluates the combination CLSC systems which incorporate English and Chinese systems. As can be seen in Table 2, the combination CLSC systems further improve the English and Chinese systems performances. In Book, DVD and Music categories, the combination CLSC systems improve 1.53%, 1.40% and 2.93% respectively, compared to the better one of English and Chinese systems. The results illustrate that the combination CLSC system increases the possibility of a review being correctly predicted. The combination of English and Chinese systems could effectively eliminate the gap between the two languages.

Table 2. Performance of combination CLSC systems

System	Book	DVD	Music	Accuracy
English system	73.03%	76.93%	75.15%	75.04%
Chinese system	78.15%	75.05%	74.30%	75.83%
Combination system	79.68%	78.33%	78.08%	78.70%

4.3 Effect of Destruction Fraction in Denoising Autoencoders

Fig. 5 shows the accuracy curve of the CLSC systems with different destruction fractions. The destruction fraction ν varies from 0 to 0.9. When ν is set to 0, denoising autoencoders are degenerated into autoencoders. As can be seen from the figure,

denoising autoencoders outperform autoencoders from $\nu=0.2$ to $\nu=0.5$. We can conclude that adding noises to the training examples properly enhances the robustness to the translation noises in the CLSC task. Denoising autoencoders with $\nu=0.2$ achieve the highest average accuracy 80.02%, which surpass the original autoencoders by 0.94%.

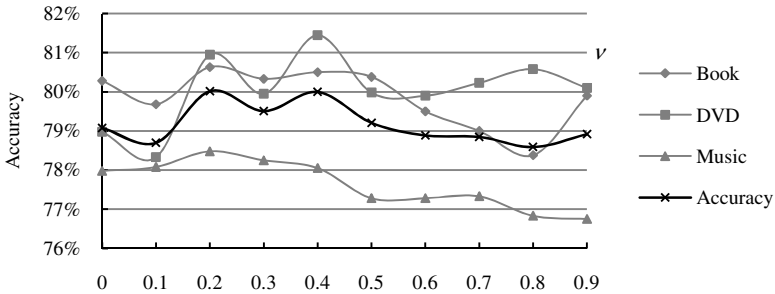


Fig. 5. Accuracy vs. Destruction fraction

4.4 Comparison with Related Work

Table 3 summarizes recent results on NLP&CC 2013 CLSC dataset. Chen et al. [10] gave different weights to sentiment words according to emotional color difference of sentiment words in subject-predicate component. In NLP&CC 2013 CLSC evaluation, they achieved the second place of accuracy. The top performer of NLP&CC 2013 CLSC share task is HLT-Hitsz system, which obtained an accuracy 77.12% by using the co-training model. They extended their model by incorporating transfer self-training model and co-training model [8], and achieved 78.89% accuracy. These methods are all based on shallow learning algorithms. We build a deep architecture to discover intermediate representation of features and achieve 80.02% accuracy.

Table 3. CLSC performance comparison on the NLP&CC 2013 Share Task test data

Team	Book	DVD	Music	Accuracy
Chen et al. [10]	77.00%	78.33%	75.95%	77.09%
HLT-Hitsz	78.50%	77.73%	75.13%	77.12%
Gui et al. [8]	78.70%	79.65%	78.30%	78.89%
Our Approach	80.63%	80.95%	78.48%	80.02%

5 Conclusion and Future Work

This paper proposes a combination CLSC approach based on denoising autoencoder to eliminate the language gap between English and Chinese. Experimental results on NLP&CC 2013 CLSC dataset show that both of the denoising autoencoder and

combination approach could improve the sentiment classification performance. In addition, we show that the feature setting of CHI feature selection method together with TF-IDF weight calculation method works well on CLSC task.

This work only combines English and Chinese systems linearly, and it's very likely that better performance could be achieved by deep combination, such as co-training and transfer learning, etc. We leave that as a future work. Meanwhile, we will select high-quality translated reviews to further reduce the impacts of translation errors.

Acknowledgements. This research is supported by Natural Science Foundation of China (No. 61272375, No. 61173100, and No. 61173101).

References

1. Turney, P.D.: Thumbs up or thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistic, pp. 417–424. Association for Computational Linguistics (2002)
2. Wei, B., Pal, C.: Cross lingual adaptation: an experiment on sentiment classifications. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 258–262. Association for Computational Linguistics (2010)
3. Zhao, Y.Y., Qin, B., Liu, T.: Sentiment Analysis (in Chinese). Journal of Software 21(8), 1834–1848 (2010)
4. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of the ACL-2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. Association for Computational Linguistics (2002)
5. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Computational Intelligence 22(2), 110–125 (2006)
6. Li, S., Xia, R., Zong, C.Q., Huang, C.R.: A framework of feature selection methods for text categorization. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJFNLP of the AFNLP, pp. 692–700. ACL and AFNLP (2009)
7. Wan, X.J.: Co-training for cross-lingual sentiment classification. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, vol. 1, pp. 235–243. Association for Computational Linguistics (2009)
8. Gui, L., Xu, R., Xu, J., Yuan, L., Yao, Y., Zhou, J., Qiu, Q., Wang, S., Wong, K.-F., Cheung, R.: A Mixed Model for Cross Lingual Opinion Analysis. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) NLPCC 2013. CCIS, vol. 400, pp. 93–104. Springer, Heidelberg (2013)
9. Li, S., Wang, R., Liu, H., Huang, C.-R.: Active Learning for Cross-Lingual Sentiment Classification. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) NLPCC 2013. CCIS, vol. 400, pp. 236–246. Springer, Heidelberg (2013)
10. Chen, Q., He, Y.X., Liu, X.L., Sun, S.T., Peng, M., Li, F.: Cross-Language Sentiment Analysis Based on Parser. Acta Scientiarum Naturalium Universitatis Pekinensis 50(1), 55–60 (2014) (in Chinese)
11. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2, 1–127 (2009)

12. Bengio, Y., Delalleau, O.: On the expressive power of deep architectures. In: Kivinen, J., Szepesvári, C., Ukkonen, E., Zeugmann, T. (eds.) ALT 2011. LNCS, vol. 6925, pp. 18–36. Springer, Heidelberg (2011)
13. Bengio, Y., LeCun, Y.: Scaling learning algorithms towards AI. *Large-Scale Kernel Machines* 34, 1–41 (2007)
14. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. *Science* 313(5786), 504–507 (2006)
15. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems* 19, 153 (2007)
16. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103. ACM, New York (2008)
17. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8), 1915–1929 (2013)
18. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 20(1), 30–42 (2012)
19. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12, 2493–2537 (2011)
20. Tang, D., Qin, B., Liu, T., Li, Z.: Learning Sentence Representation for Emotion Classification on Microblogs. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) *NLPCC 2013*. CCIS, vol. 400, pp. 212–223. Springer, Heidelberg (2013)
21. Zhou, S.S., Chen, Q.C., Wang, X.L.: Active deep networks for semi-supervised sentiment classification. In: *COLING 2010 Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1515–1523. Association for Computational Linguistics (2010)
22. Galavotti, L., Sebastiani, F., Simi, M.: Feature selection and negative evidence in automated text categorization. In: *Proceedings of KDD (2000)*
23. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information processing & management* 24(5), 513–523 (1988)
24. Bergstra, J., Breuleux, O., Bastien, F., et al.: Theano: a CPU and GPU math expression compiler. In: *Proceedings of the Python for Scientific Computing Conference, SciPy (2010)*

Aspect-Object Alignment Using Integer Linear Programming

Yanyan Zhao¹, Bing Qin², and Ting Liu²

¹ Department of Media Technology and Art, Harbin Institute of Technology

² Department of Computer Science and Technology, Harbin Institute of Technology
{yyzhao, bqin, tliu}@ir.hit.edu.cn

Abstract. Target extraction is an important task in opinion mining, in which a complete target consists of an aspect and its corresponding object. However, previous work always simply considers the aspect as the target and ignores an important element “object.” Thus the incomplete target is of limited use for practical applications. This paper proposes a novel and important sentiment analysis task: aspect-object alignment, which aims to obtain the correct corresponding object for each aspect, to solve the “object ignoring” problem. We design a two-step framework for this task. We first provide an aspect-object alignment classifier that incorporates three sets of features. However, the objects assigned to aspects in a sentence often contradict each other. To solve this problem, we impose two kinds of constraints: intra-sentence constraints and inter-sentence constraints, which are encoded as linear formulations and use Integer Linear Programming (ILP) as an inference procedure to obtain a final global decision in the second step. The experiments on the corpora of camera domain show the effectiveness of the framework.

Keywords: Opinion Mining, Aspect-Object Alignment, Integer Linear Programming.

1 Introduction

Opinion mining and sentiment analysis entail a number of interesting and challenging tasks, such as sentiment classification, sentiment extraction and sentiment summarization and so on. A fundamental one is target extraction, which aims to recognize the main topic that has been commented on in a review. Generally speaking, the target is composed of object and aspect. For example, in the review “I bought [*Canon 600D*]^o yesterday, its [*photos*]^a are amazing,” the complete target is $\langle \textit{photos}, \textit{Canon 600D} \rangle$. However, previous work always considers only the aspect as target [5,16], in which the target is not complete. For instance, in the aforementioned review, the aspect “*photos*” in the second sentence is always directly tagged as the target. Apparently, this target is incomplete because it ignores a very important element, which is the object “*Canon 600D*” that the aspect “*photos*” belongs to. We call this problem as “object ignoring”. The incomplete target is of limited use for practical applications if we do not recognize the object.

In this paper, we define the target consist of two parts, namely, the aspect and its corresponding object, such as $\langle photos, Canon\ 600D \rangle$. Correspondingly, the task of target extraction can be composed of two main subtasks:

- Aspect/object extraction, which aims to extract the aspects/objects in sentiment sentences.
- Find the correct, corresponding object for each aspect in the review. In this paper, we call this task as ***Aspect-Object Alignment***.

Currently, researchers mainly focus on the first task and ignore the second task even though this task is more important for practical applications. The target distribution statistics show that only 10% aspects can explore its objects in the same sentence, which is very low. This result illustrates that just a few aspects and their corresponding objects are co-occurring in the same sentence, such as them in the sentence “the $[appearance]^a$ of $[GF3]^o$ is very beautiful.” However, most of the aspects and their corresponding objects do not co-occur, such as them in “the $[appearance]^a$ is very beautiful.” Therefore, we should explore the objects for most aspects in other sentences and choose the right one from several object candidates. All the above can indicate that aspect-object alignment is a new and necessary task. As such, this paper mainly focuses on this task. We hypothesize that all aspects and objects appearing in each sentiment sentence have been manually annotated.

We can simply regard the aspect-object alignment task as a binary classification, in which each decision is made in a pair-wise manner made of aspect and object, independently of others. Moreover, we propose three sets of features for the aspect-object classifier, namely, the basic, relational, and special target features. However, there is one major drawback with this method. That is, it makes each decision independently of previous ones in a greedy way. Clearly, the determination of the relation between the aspect and object should be conditioned on how well it works as a whole.

We tackle this issue by recasting the task of aspect-object alignment as an optimization problem, namely, an Integer Linear Programming (ILP) problem. ILP can perform global inference based on the output of the classifier, which makes it highly suitable to address the aforementioned problem. ILP-based models have been developed for many tasks that range from semantic role labeling [11] to multi-document summarization [10], and opinion mining [1]. In this paper, we firstly use ILP to search for a global assignment based on decisions obtained through the binary aspect-object classifier alone. Second, we provide the joint formulation, in which constraints are added to ensure that the ultimate results are mutually consistent.

We construct two kinds of constraints.

Intra-Sentence Constraints: They describe the constraints between objects and aspects or between two aspects, where the objects and aspects appear *in a same sentiment sentence*. For example, if a normal sentence (non-comparative sentence) contains two aspects but no object, then the two aspects share the same object.

Inter-Sentence Constraints: They describe the constraints between objects and aspects or between two aspects, where the objects and aspects appear *in different sentiment sentences*. This kind of constraint always uses the idea of sentiment consistency, which is inspired by the work of Ding et al. [3].

We evaluate our proposed framework on a review corpus of digital camera domain. The experimental results show that the two kinds of constraints achieve significant performances, which are higher than that of the base classifier. The joint model particularly achieves accuracy improvements of more than 5% over the cascading rule-based baseline and nearly 2% over the aspect-object alignment classifier.

2 Related Work

Target extraction is an important task in sentiment analysis, which has recently attracted much attention. Many efficient approaches have been developed for this task, which can be divided into three kinds of methods, namely, rule-based [5,16,12], supervised [14,6], and topic model-based [9,7,8] methods. However, most researchers consider the aspects alone as the targets; thus, the mentioned algorithms are all designed for aspect extraction. The “object” element in the target is ignored, and few researchers are studying on the aspect-object alignment task.

The aspect-object alignment task is similar to the entity assignment task [3], which assigns objects to each sentence in a review. Many rules have been proposed to simply solve this task, which mainly focus on processing the comparative sentences by using the idea of sentiment consistency. The major problem is the sequential application of the rules in this method, which sometimes causes conflicts. Thus, this method cannot effectively achieve an optimal result. This task is actually different from our proposed study, which aims to find an object for each aspect. However, we can simply modify this method as a baseline and use it as a comparative method for our approach.

Aspect-object alignment task is also similar to the task of coreference resolution [13,4] to a particular extent, which has been considerably studied previously. In this task, aspect can be treated as anaphor, and object can be treated as entity. Recently, several researchers studied coreference resolution for the review texts [2]. We can observe that the most significant approach is based on supervised learning, in which a pair-wise function is used to predict a coreferent pair of noun phrases. These methods are an inspiration for us to learn several useful features, such as distance features, for the aspect-object alignment task.

3 Method

This section introduces the proposed overall framework. The framework consists of two main steps, which are learning an aspect-object alignment classifier and using an ILP inference. Generally speaking, an aspect-object alignment classifier

is learned to estimate the probability for each pair of aspect and object. We use an ILP inference procedure to achieve an optimal global result by considering specific special constraints.

3.1 Aspect-Object Alignment Classifier

We can generate the Cartesian product of all the aspects and objects in each review into a pair-wise vector $\langle a, o \rangle$. The aspect-object alignment task can be converted to classify each $\langle a, o \rangle$ into true or false. We use a maximum entropy model for the aspect-object alignment classifier.

The features are generated from three kinds of relative sentences, which are shown as follows. Figure 1 shows the examples.

Review1:
s1: [Canon S100]^o is really good for general use.
s2: I bought one last year.
s3: The [screen]^a is really clear.

Review2:
s1: The [screen]^a is really clear.
s2: I strongly recommend [Canon S100]^o !

Fig. 1. Example of three kinds of sentences

- Present sentence: this kind of sentence refers to the sentence containing a , such as s_3 of review1 and s_1 of review2 in Figure 1.
- Previous sentence: this kind of sentence should satisfy two conditions: (1) it contains objects and; (2) it is the nearest previous sentence to the present sentence. For example, s_1 of review1 is the previous sentence of the present sentence s_3 . We explore the features generated from this kind of previous sentence, because in particular cases, the corresponding object for a in the present sentence can be acquired from the previous sentence.
- Nearest sentence: this kind of sentence should also satisfy two conditions: (1) it contains objects, and (2) it is nearest to the present sentence. Note that, the nearest sentence is different from the previous sentence, because sometimes it can be found in the sentences after the present sentence. We consider the features generated from the nearest sentence because in particular cases, the corresponding object of a in the present sentence can be acquired from the nearest sentence.

Based on the three kinds of sentences, we propose three categories of features, which are basic, relational and special target features, as follows:

Basic Features: We design several basic features from the present, previous and nearest sentence respectively.

- **Sentence Type Feature:** Sentiment sentences can be divided into three types based on the objects or aspects it contains. The first one contains

objects but no aspect. For example, “[*Canon.S100*]^o is really good for general use.” The second one contains objects and aspects at the same time. For example, “The [*screen*]^a of [*Canon.S100*]^o is really clear.” The third one contains aspects but no object. For example, “The [*screen*]^a is really clear.” Their possible encoded values are 01, 02 and 03 respectively. This feature is used to describe the present/previous/nearest sentence respectively. We design this feature because the methods of aspect-object alignment vary for different types of sentences.

- **Comparative Sentence Feature:** Sentiment sentences can be divided into normal and comparative sentences. If the present/previous/nearest sentence is a comparative sentence, the value is true; otherwise, is false. We design this feature because the method of aspect-object alignment for normal sentences is different from the method for comparative sentences.
- **Object Feature:** This feature refers to the object that appears in the present/previous/nearest sentence.

Relational Features: We also design several relational features. One is the distance feature, which is inspired by the coreference resolution task [2]. The other one is object consistency feature, which is inspired by Ding et al. [3].

- **Distance between present and previous sentence:** The possible values are 0, 1, 2, 3 and so forth, which captures the sentence numbers between the present and its previous sentence.
- **Distance between present and nearest sentence:** The possible values are 0, 1, 2, 3 and so forth, which captures the sentence numbers between the present and its nearest sentence.
- **Consistency between the object in previous sentence and the candidate object in $\langle a, o \rangle$:** If the object in the previous sentence is the same as the one in $\langle a, o \rangle$, the value is true; otherwise, the value is false.
- **Consistency between the object in nearest sentence and the candidate object in $\langle a, o \rangle$:** If the object in the nearest sentence is the same as the one in $\langle a, o \rangle$, the value is true; otherwise, the value is false.

Special Target Features

- **First appearing object in the review:** This feature refers to the object that appears for the first time in the review.
- **Most frequent object in the review:** This feature refers to the object that appears most frequently in the review.

Based on the above features, we can build an aspect-object alignment classifier to judge each $\langle a, o \rangle$ candidate.

3.2 ILP Inference

In an ideal setting that has a perfect aspect-object alignment classifier, each aspect can obtain the correct object according to the classifier’s prediction.

In reality, labels (objects) assigned to aspects in a sentence often contradict each other. For example, each aspect has only one object. These complicated features are difficult to use in the classifier. Therefore, we encode the constraints as linear formulations to resolve the conflicts. We also use ILP as an inference procedure to make a final decision that is consistent with the constraints.

We formally define the aspect set as $A = \{a_1, a_2, \dots, a_n\}$, and the object set as $O = \{o_1, o_2, \dots, o_m\}$ for each review. We assume that the resulting object set for the aspects in A is $S = \{s_1, s_2, \dots, s_n\}$, and $s_i \in O$, which is the resulting object for a_i . Thus, this task can be formulated such that the aspect-object assignment classifier attempts to assign object from O for each a_i in set A . If we assume that the classifier returns a probability value, $p(a_i, s_i)$, which corresponds to the likelihood of assigning label s_i for aspect a_i , then the inference task in a given review can be depicted by maximizing the overall score of the aspects as follows. Moreover, \hat{S} is the optimal result among all the potential vectors.

$$\hat{S} = \operatorname{argmax} \sum_{i=1}^n p(a_i, s_i) \quad (1)$$

In this equation, the probability $p(a_i, s_i)$ can be obtained through the aforementioned aspect-object alignment classifier in Section 3.1. If s_i is the j^{th} object in the object set O , then $p(a_i, s_i)$ can be denoted as p_{ij} , which represents the probability of the pair of the i^{th} aspect in A and j^{th} object in O .

We can introduce a set of binary indicator variables $z_{ij} \in \{0, 1\}$ for each p_{ij} to reformulate the original \hat{S} function into a linear function, to acquire the optimal \hat{S} . If the resulting corresponding object for a_i is actually o_j , then the value $z_{ij} = 1$; otherwise, $z_{ij} = 0$. Thus, the task of finding an optimal \hat{S} can be converted to find an optimal vector Z that can maximize the objective function. Here, Z is the set of z_{ij} , and $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, m\}$. The equation (2) can be written as an ILP objective function as follows:

$$\hat{Z} = \operatorname{argmax} \sum_{i=1}^n \sum_{j=1}^m p_{ij} z_{ij} \quad (2)$$

Subject to

$$\sum_{j=1}^m z_{ij} = 1, \quad \forall z_{ij} \in Z \quad (3)$$

Note that although this constraint comes from the variable transformation, it has a real meaning, which denotes that each aspect can take only one object. This constraint can be considered as **constraint 1**.

Next, we impose several constraints on equation (3) to acquire the optimum solution. The designed constraints can be divided into two categories, which are **intra-sentence constraints** and **inter-sentence constraints** defined in Section 1.

Intra-sentence Constraints. Constraint 2 to Constraint 4 are the representatives.

Constraint 2: If the aspect a_p and a_q are in the same sentence s , and no object appears in the sentence, then a_p has the same object as a_q has. For example, in the sentence “It has a good [*resolution*]^a and a good [*LCD screen*]^a,” “*resolution*” and “*LCD screen*” satisfy this constraint; thus, they have the same object. This constraint can be represented using the following equation.

$$\forall j \in \{1, \dots, m\} : z_{pj} = z_{qj}, \quad (4)$$

where two aspects a_p and a_q , but no object appear in s .

Constraint 3: This constraint is available only when the given sentence s can satisfy three conditions: (1) s is a comparative sentence; (2) s contains two objects o_k and o_t , which appear on both sides of the comparative word; and (3) only one aspect a_p appears in this sentence. Then the corresponding object for a_p is one of the two appearing objects in the given sentence. For example, in the sentence “the [*shutter sound*]^a of [*Canon 5D3*]^o is better than [*5D2*]^o’s,” the aspect “*shutter sound*” definitely belongs to one of the objects appearing in this sentence. The object in this sentence is “*Canon 5D3*.” This constraint can be described by the following formulation:

$$z_{pk} + z_{pt} = 1, \quad (5)$$

where only one aspect a_p and two objects o_k , o_t appear in the comparative sentence s ; and meanwhile o_k , o_t appear on both sides of the comparative word.

Constraint 4: This constraint is available only when the given sentence s can satisfy two conditions: (1) s contains only one aspect a_p and one object o_k ; and (2) this sentence is a normal sentence. Then the corresponding object for a_p is the object o_k . For example, in the sentence “the [*shutter sound*]^a of [*Canon 5D3*]^o is good,” “*Canon 5D3*” is the right object for the aspect “*shutter sound*.” This constraint can be described by the following formulation:

$$z_{pk} = 1, \quad (6)$$

where only one aspect a_p and one object o_k appear in the normal sentence s .

Inter-Sentence Constraints. Many previous research illustrate that adjacent sentences in a review have particular sentiment relationships [15,3]. Thus, the sentiment orientations for two adjacent sentences are always the same or totally different (because of the usage of transitional words, such as “but”), which is named as “sentiment consistency.” This idea is also very useful for the aspect-object alignment task. We design two constrains based on this idea as follows.

Constraint 5: This constraint is available only when the given sentence s_g can satisfy three conditions: (1) s_g only contains an aspect a_p , but no object; and (2)

the previous sentence s_p is a normal sentence, which contains an aspect a' and an object o_k ; and (3) the sentiment orientation of s_g is the same as that of s_p . Then the corresponding object for a_p is the object o_k in the previous sentence s_p . For example, in the sentence “The best feature about [*Canon S110*]^o is its [*size*]^a. The [*picture quality*]^a is good, too.”, the second sentence satisfies this constraint, and the corresponding object for the aspect “*picture quality*” is the object “*Canon S110*” in the previous sentence. This constraint can be described by the following formulation:

$$z_{pk} = 1, \tag{7}$$

where only one aspect a_p but no object appear in the given sentence s_g ; and only one aspect a' and one object o_k appear in the normal previous sentence s_p ; meanwhile, $polarity(s_g) = polarity(s_p)$.

Constraint 6: This constraint is available only when the given sentence can satisfy two conditions: (1) the given sentence s_g contains an aspect a_p , but no object; and (2) the previous sentence s_p is a comparative sentence, and contains two objects o_k and o_t , which appear on both sides of the comparative word and o_k is in front of o_t . If the given sentence shows the same sentiment orientation as the previous sentence, then the corresponding object for a_p is the object o_k in the previous sentence. However, if the given sentence shows a different sentiment orientation with the previous sentence, the object for a_p is o_t .

For example, in the review “the [*shutter sound*]^a of [*Canon 5D3*]^o is better than [*5D2*]^os. The [*picture quality*]^a is good, too.”, the object for “*picture quality*” in the second sentence is “*Canon 5D3*” in the previous sentence, but not “*5D2*.” This constraint can be described by the following formulation:

$$\begin{aligned} z_{pk} &= 1, & \text{if } polarity(s_g) &= polarity(s_p) & (8) \\ z_{pt} &= 1, & \text{if } polarity(s_g) &= -polarity(s_p), & (9) \end{aligned}$$

where only one aspect a_p but no object appear in the given sentence s_g ; and two objects o_k and o_t appear in the previous comparative sentence s_p and on both sides of the comparative word.

The intra-sentence and inter-sentence relationships discussed in the previous sections can be encoded as constraints in an ILP inference process. By formulating the problem this way, we can use the aspect-object alignment classifier and also jointly use an ILP model and many useful constraints to generate the optimal results.

4 Experimental Setup

Corpus. We manually collected on-line customer reviews of digital camera as a case study for the aspect-object alignment task. The corpus is from two famous Chinese forum sites, namely, <http://ww.xitek.com/> and <http://www.fengniao.com/>. The statistics of the corpus are illustrated in Table 1.

Table 1. Statistics for the corpus

No.	Types	Digital camera
1	# of reviews	200
2	# of sentences	8,042
3	# of aspects	2,017
4	Average # of object for each review	2.82
5	# of pairwise $\langle a, o \rangle$	9,161

The raw corpus has 200 documents, in which 8,042 sentences are annotated. We summarized some statistics, which shows 2,017 aspects in the corpus and an average of 2.82 objects for each review. According to the Cartesian product of all aspects and objects, each review in the corpus has a total of 9,161 aspect-object pairs, which are also the input of the binary aspect-object alignment classifier.

Baselines. We compare our system with two baselines. **Baseline1** is a cascading rule-based approach, which is similar to the method of Ding et al. [3]. This approach combines several useful rules including the idea of sentiment consistency, but several rules are conflicting with one another during processing. **Baseline2** is the aspect-object alignment classifier without the ILP inference.

Training and Evaluation. We experiment with ME algorithm and tune the related parameters for the aspect-object alignment classifier by using the 10-fold cross-validation on the corpus described in the previous section. Because our goal of the aspect-object alignment task is to find a certain object for each aspect, the value *Accuracy* is suitable for this task.

5 Results and Discussion

5.1 Results of Our Method and Two Baselines

Table 2 shows the performances of our ILP inference method and two baselines, the cascading rule-based approach, and aspect-object alignment classifier.

Table 2. Comparative results of our method and two baselines

Method	<i>Accuracy</i> (%)
Baseline1: cascading rule-based	78.04
Baseline2: aspect-object alignment classifier	81.80
Our ILP inference method	83.69

Table 2 shows that the performance of the cascading rule-base approach is not very ideal. The reason is attributed to its nature as a rule based method, where the rules are used sequentially and sometimes have conflicts in them. Baseline2, which is the aspect-object alignment classifier, achieves an accuracy of 81.80%,

which significantly (χ test with $p < 0.0001$) outperforms the cascading rule-based approach. The features proposed in this method, which can globally describe this task, provide a major difference.

In addition, our method, which combines an aspect-object alignment classifier and an ILP inference processing, performs best and significantly (χ test with $p < 0.03$) outperforms the two baselines. We can obviously observe that our method with ILP inference performs better than the method (namely, baseline2) without it. This illustrates that the ILP inference is useful, and the two kinds of constraints designed in this paper are effective.

5.2 Aspect-Object Alignment with ILP Inference

ILP is used to perform global inference based on the classifier’s output to resolve conflicts between rules. Six constraints are used in this paper, in which Constraint 1 to Constraint 4 reflect the constraints within a sentiment sentence, whereas Constraint 5 and 6 reflect the constraints between different sentiment sentences. Table 3 lists the performance of the system with different constraints.

Table 3. Results of aspect-object alignment with different ILP constraints

Constraints	ILP constraints	Accuracy (%)
Intra-sentence constraints	ILP-c1	81.80
	ILP-c2	81.85
	ILP-c3	81.90
	ILP-c4	82.65
Inter-sentence constraints	ILP-c5	82.45
	ILP-c6	82.05
All constraints	ILP-c1-6	83.69

Table 3 shows that:

- Constraint 1 refers that there is only one object for each aspect, thus the result with this constraint is equivalent to the basic aspect-object alignment classifier.
- Constraint 2 to Constraint 4 are intra-sentence constraints, and align the aspect-object pair according to the structural information (such as the relationships between aspects and objects) conveyed from the given sentence. The improvement of Constraint 4 among all the constraints is apparent, which illustrates the effectiveness of this constraint. Constraint 2 and 3 can slightly increase the accuracy compared with the basic classifier without ILP inference. That’s because most of the instances that satisfy these two constraints can also obtain the correct results by using the basic classifier.
- Constraint 5 and 6 are inter-sentence constraints, and mainly use the idea of sentiment consistency. Table 3 shows that the improvement of Constraint

5 and Constraint 6 are all apparent. These two constraints primarily focus on sentences that contain aspects but no objects. The statistics show that the proportion of this kind of sentences is about 90%, which is very high. The effectiveness of these two constraints demonstrates that for this kind of sentences, we can seek for the corresponding objects in their neighboring sentiment sentences. Moreover, this can further demonstrate that sentiment consistency can improve the task’s performance.

- We combine Constraint 1 to Constraint 6 as the final inference and obtain the best performance of 83.69%. It can further improve the aspect-object alignment task by about 2%, which significantly (χ test with $p < 0.03$) outperforms the aspect-object alignment classifier without ILP inference.

6 Conclusion and Future Work

In this paper, we propose a novel and important sentiment analysis task, *aspect-object alignment*, which aims to resolve the “object ignoring” problem in target extraction. We propose a two-step framework for this task, including an aspect-object alignment classifier and an ILP inference. The experimental results show that the aspect-object alignment classifier with the ILP inference performs better than the classifier without it and also illustrate that the six constraints proposed in this paper are very useful.

In the future, we will endeavor to enhance the algorithm for each step by seeking for more useful features for the classifier or more useful constraints for the ILP inference to improve the performance of the task.

Acknowledgments. We thank the anonymous reviewers for their helpful comments. This work was supported by National Natural Science Foundation of China (NSFC) via grant 61300113, 61133012 and 61273321, and the Ministry of Education Research of Social Sciences Youth funded projects via grant 12YJCZH304.

References

1. Choi, Y., Cardie, C.: Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 2, pp. 590–598 (2009)
2. Ding, X., Liu, B.: Resolving object and attribute coreference in opinion mining. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 268–276. Association for Computational Linguistics, Stroudsburg (2010)
3. Ding, X., Liu, B., Zhang, L.: Entity discovery and assignment for opinion mining applications. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 1125–1134. ACM, New York (2009)

4. Gärtner, M., Björkelund, A., Thiele, G., Seeker, W., Kuhn, J.: Visualization, search, and error analysis for coreference annotations. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 7–12 (June 2014)
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of KDD 2004, pp. 168–177 (2004)
6. Jakob, N., Gurevych, I.: Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 1035–1045. Association for Computational Linguistics, Stroudsburg (2010)
7. Liu, K., Xu, L., Zhao, J.: Extracting opinion targets and opinion words from online reviews with graph co-ranking. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 314–324 (June 2014)
8. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: Modeling facets and opinions in weblogs. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 171–180 (2007)
9. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 339–348. Association for Computational Linguistics, Jeju Island (2012)
10. Nishikawa, H., Hasegawa, T., Matsuo, Y., Kikui, G.: Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010, pp. 910–918. Association for Computational Linguistics, Stroudsburg (2010)
11. Punyakanok, V., Roth, D., Yih, W.T., Zimak, D.: Semantic role labeling via integer linear programming inference. In: Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, Association for Computational Linguistics, Stroudsburg (2004)
12. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37(1), 9–27 (2011)
13. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27(4), 521–544 (2001)
14. Yu, J., Zha, Z.J., Wang, M., Chua, T.S.: Aspect ranking: Identifying important product aspects from online consumer reviews. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 1496–1505. Association for Computational Linguistics, Stroudsburg (2011)
15. Zhao, J., Liu, K., Wang, G.: Adding redundant features for crfs-based sentence sentiment classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 117–126. Association for Computational Linguistics, Stroudsburg (2008)
16. Zhao, Y., Qin, B., Hu, S., Liu, T.: Generalizing syntactic structures for product attribute candidate extraction. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 377–380. Association for Computational Linguistics, Los Angeles (2010)

Sentiment Classification of Chinese Contrast Sentences

Junjie Li¹, Yu Zhou¹, Chunyang Liu², and Lin Pang²

¹ National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

² National Computer Network Emergency Response Technical
Team/Coordination Center of China, Beijing, 100029
{junjie.li,yzhou}@nlpr.ia.ac.cn,
lcy@isc.org.cn, panglin_cncert@163.com

Abstract. We present the study of sentiment classification of Chinese contrast sentences in this paper, which are one of the commonly used language constructs in text. In a typical review, there are at least around 6% of such sentences. Due to the complex contrast phenomenon, it is hard to use the traditional bag-of-words to model such sentences. In this paper, we propose a Two-Layer Logistic Regression (TLLR) model to leverage such relationship in sentiment classification. According to different connectives, our model can treat different clauses differently in sentiment classification. Experimental results show that TLLR model can effectively improve the performance of sentiment classification of Chinese contrast sentences.

Keywords: Two-Layer Logistic Regression (TLLR) Model, sentiment classification.

1 Introduction

Sentiment analysis is an active research area and its main job is to identify the attitude of a text, a sentence or a phrase[1,6,10]. Both lexicon-based [4,12] and corpus-based [8,9] approaches exist for this task. In the corpus-based methods, the most common one for sentiment classification is the statistical model based on bag-of-words (BOW) [8]. Although the method is simple and effective, there are still many shortcomings, e.g., BOW ignores the relationship among words and that among sentences. Due to these defects, BOW method cannot handle many useful linguistic phenomena for sentiment classification, such as the contrast.

On one hand, in order to incorporate the relationship among words, [3] used contextual clues to disambiguate polarity. [14] proposed a dual training and dual prediction method to deal with the negation phenomenon.

On the other hand, in order to incorporate the relation among sentences, many researchers [2,11,13] make use of discourse structure to assist sentiment classification. Their experimental results show that the contrast relation is very important for sentiment classification. According to the discourse theory, a contrast sentence usually contains two clauses: the nuclei clause and the satellite clause. Previous works

concentrate on treating these two parts differently in sentiment classification. [2,11,13] think different clauses in the contrast sentence have different functions in expressing sentiments. Their experiment results show that the nuclei clause is more important than the satellite one in the task. [7] studies sentiment analysis of conditional sentence systematically. [5] considers the contrast phenomenon as one kind of polarity shifting. Through splitting the sentence into polarity-shifted parts and polarity-unshifted parts, they designed three strategies (remove, shift and joint) to deal with the different parts. The most effective strategy is the joint strategy, which uses different BOW models for the two parts and learns the different BOW weights together and their experiment results support their method.

However, there are still two shortcomings in the previous works: Firstly, it is not adequate to assign different weights to the satellite and nuclei clause in contrast relation. Let's observe the following two example sentences (1) and (2).

- (1) 酒店位置好|The hotel location is good, 只是设施糟糕|yet its facilities are poor.
 (2) 酒店位置好|The hotel location is good, 但是设施糟糕|but its facilities are poor.

Except connectives, the two sentences consist of same words but they have opposite polarity. The reason is that different connectives in the same relation have different tendency degrees. The word “只是|yet” is a slight contrast connective. When two clauses have different polarities, the word usually makes the sentiment tend to the polarity of the former clause. However, the conclusion is different from the connective “但是|but”, which always tends to polarity of the latter clause. Therefore, connectives are crucial for sentiment analysis, which are totally ignored in previous works.

Secondly, all the existing methods work as a pipeline schema. They learn word-level weights and clause-level weights separately. The pipeline approach often causes error propagation. Wrong word-level weights in the earlier stage harm the subsequent sentence sentiment label. To our best knowledge, there is no work to train word-level weights and clause-level weights jointly.

To address the issues mentioned above, we propose a Two-Layer Logistic Regression (TLLR) model in this paper, which jointly learn weights in both clause-level and word-level and consider the element of different connectives. Experimental results show that TLLR model can effectively improve the performance of sentiment classification of Chinese contrast sentences.

Although this paper only studies contrast sentences, it is also easy to integrate this method to an overall sentiment analysis or opinion mining system. The reason is that it is easy to identify contrast sentences using connectives in Table 1. When we find the input sentence is a contrast sentence, we can use the method in this paper to get the polarity.

2 Approach Overview

In this paper, we only focus on contrast sentences, which have explicit connectives (in Table 1) to split the sentence into different parts. Given such a sentence, Figure 1

depicts the general procedure of our approach. We first use connectives (in Table 1) to split the sentence into the nuclei clause and satellite clause, then the TLLR model works on the structure and returns the sentiment polarity of the sentence.

Table 1. Explicit connectives in contrast relation

Relation	Connectives
转折	不过 however 但是 but
contrast	但 but 只是 yet 可是 however

The nuclei clause in a contrast sentence is the clause after the connective, while the satellite clause is the clause before the connective. From the following example, we can get more insights.

Input: 酒店位置好|The hotel location is good, 但是设施糟糕|but its facilities are poor.

Output: Connective: “但是|but”, Relation: “转折|Contrast”. Satellite Part: “酒店位置好|The hotel location is good”, Nuclei Part: “只是设施糟糕|but its facilities are poor”.

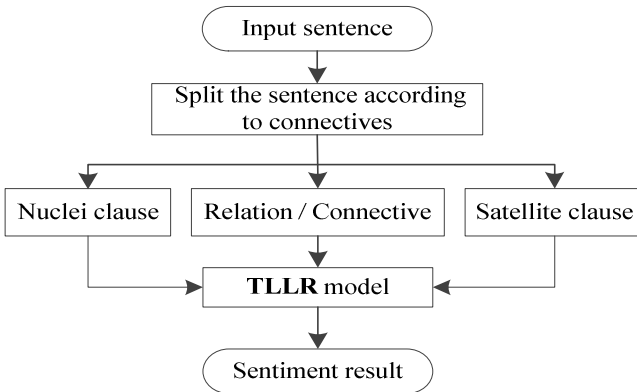


Fig. 1. Our approach overview

3 Two-Layer Logistic Regression Model

Logistic regression (LR) is a very famous model for two-class classification, which can deal with sentiment classification with BOW features. However, when we consider the contrast relation in sentiment classification, the traditional LR model based on BOW cannot handle the relation. Therefore, a Two-Layer Logistic Regression (TLLR) model is proposed to remedy the defect of LR model.

3.1 Logistic Regression Model

The structure of LR model is shown in Figure 2. The input for LR model is feature vector (\vec{x}) and the parameter is the weight vector ($\vec{\theta}$). Category (y) is the output

corresponding to the feature vector. Sigmoid function is utilized to map the dot product of feature vector and weight vector into category. The mathematical formula about $\vec{x}, \vec{\theta}, y$ can be seen in Equation (1) below.

$$P(y = 1 | \vec{x}) = h(\vec{\theta} \cdot \vec{x}) = \frac{1}{1 + e^{-\vec{\theta} \cdot \vec{x}}} \quad (1)$$

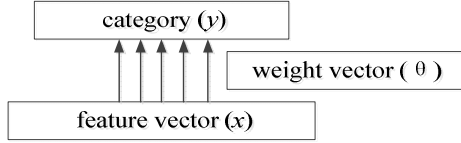


Fig. 2. The structure of Logistic Regression model

Loss Function: Suppose we have a dataset with N samples, the k -th sample has feature vector (\vec{x}_k) , category (y_k) . The loss function for the k -th sample is shown in Equation (2).

$$Cost(h(\vec{\theta} \cdot \vec{x}_k), y_k) = \begin{cases} -\log(h(\vec{\theta} \cdot \vec{x}_k)) & \text{if } y_k = 1 \\ -\log(1 - h(\vec{\theta} \cdot \vec{x}_k)) & \text{if } y_k = 0 \end{cases} \quad (2)$$

Equation (3) gives the whole loss function for the dataset.

$$J(\vec{\theta}) = \sum_{k=1}^N Cost(h(\vec{\theta} \cdot \vec{x}_k), y_k) \quad (3)$$

We use L-BFGS algorithm to get the parameters.

3.2 Two-Layer Logistic Regression Model

TLLR model is proposed to make use of the contrast relation to improve sentiment classification. The structure of the model is presented in Figure 3.

For a contrast sentence, we use the connective in Table 1 to split the sentence into two clauses: the satellite clause and the nuclei clause. Then two feature vectors (satellite feature vector $(\vec{x}_{sat,rel})$ and nuclei feature vector $(\vec{x}_{nu,rel})$) are employed to represent the two clauses in contrast sentence. We can get the nuclei score from the dot product of word weights vector in nuclei $(\vec{\theta}_{nu})$ and nuclei feature vector $(\vec{x}_{nu,rel})$. Similarly, we can also get the satellite score. Thus, the contrast sentence score can be computed by using the clause-level weights $(\alpha_{sat,rel}$ and $\alpha_{nu,rel})$ to combine the nuclei score and the satellite score. The sigmoid function can be employed to map the contrast sentence score into category (y) . The mathematical formula can be seen in Equation (4).

$$P(y = 1 | \vec{x}) = h\left(\alpha_{sat,rel} \times (\vec{\theta}_{sat} \cdot \vec{x}_{sat,rel}) + \alpha_{nu,rel} \times (\vec{\theta}_{nu} \cdot \vec{x}_{nu,rel})\right) \quad (4)$$

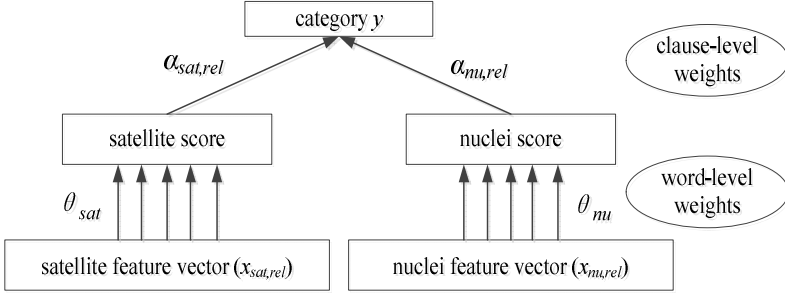


Fig. 3. The structure of Two-Layer Logistic Regression model

In Equation (4), $h(\bullet)$ is the sigmoid function. The variable sat and nu represents the satellite part and nuclei part of contrast sentences respectively. rel represents the relationship or connective in contrast sentences.

Here, we set a constraint for the nuclei and satellite part in the same rel .

$$\forall rel, \alpha_{sat,rel} + \alpha_{nu,rel} = 1 \tag{5}$$

Equation (4) is changed into Equation (6).

$$P(y = 1 | \bar{x}) = h\left(\alpha_{sat,rel} \times (\bar{\theta}_{sat} \cdot \overline{x_{sat,rel}}) + (1 - \alpha_{sat,rel}) \times (\bar{\theta}_{nu} \cdot \overline{x_{nu,rel}})\right) \tag{6}$$

Loss Function: For a dataset with N samples, the k -th sample has a feature vector (\bar{x}_k) , category (y_k) . The loss function for the k -th sample is shown in Equation (7).

$$Cost(P(y = 1 | \bar{x}_k), y_k) = \begin{cases} -\log(P(y = 1 | \bar{x}_k)) & \text{if } y_k = 1 \\ -\log(1 - P(y = 1 | \bar{x}_k)) & \text{if } y_k = 0 \end{cases} \tag{7}$$

The whole loss function for the dataset can be seen in Equation (8).

$$J(\bar{\theta}, \alpha) = \sum_{k=1}^N Cost(P(y = 1 | \bar{x}_k), y_k) \tag{8}$$

Parameter Estimation: By minimizing the loss function, we will get all parameters. To simplify the formulation, we set:

$$g(\bar{x}_k, y_k) = P(y = 1 | \bar{x}_k) - y_k$$

For the k -th sample, we get the gradient of the word weights in Equation (9) and (10).

$$\frac{\partial J(\bar{\theta}, \alpha)}{\partial \bar{\theta}_{sat}} = g(\bar{x}_k, y_k) \times \alpha_{sat,rel} \times \overline{x_{sat,rel}} \tag{9}$$

$$\frac{\partial J(\bar{\theta}, \alpha)}{\partial \bar{\theta}_{nu}} = g(\bar{x}_k, y_k) \times (1 - \alpha_{sat,rel}) \times \overline{x_{nu,rel}} \tag{10}$$

Equation (11) gives the gradient of the clause-level weights. L-BFGS is used to get the parameters from the gradient.

$$\frac{\partial J(\bar{\theta}, \alpha)}{\partial \alpha_{sat,rel}} = g(\bar{x}_k, y_k) \times (\bar{\theta}_{sat} \cdot \overline{x_{sat,rel}} - \bar{\theta}_{nu} \cdot \overline{x_{nu,rel}}) \quad (11)$$

4 Experiments

4.1 Distribution of Contrast Sentences and Connectives

We have crawled some cellphone reviews from 360buy¹. After using sentence delimiter to split the reviews into sentences, we get 693,125 sentences. By using connectives in Table 1, we can get 41,629 contrast sentences, accounting for 6% in all sentences.

Table 2. Statistics of sentences with various contrast connectives

Contrast Connectives	Sentence Number	Sentence Proportion (%)
不过 however	14,869	35.72
但是 but	11,919	28.63
但 but	9,553	22.95
只是 yet	3,047	7.32
可是 however	2,241	5.38
All	41,629	100

A further analysis of the distribution of various connectives in contrast sentences is shown in Table 2. “不过|however”, “但是|but”, and “但|but” are the top three connectives in all contrast sentences. In the following experiment, we will find different connectives have different tendency degree for the sentiment classification.

4.2 Experiment Dataset

Reviews in 360buyare scored from 1 to 5 discretely. The higher score means more positive. We label the sentiment label of a comment review based on the score. If the score is 1 or 2, we label it negative. If the score is 4 or 5, we label it positive. We reject the review whose score is 3. After using contrast connectives to get contrast sentence review, we collect 2K contrast sentences as our experiment dataset. The dataset is balanced for two categories: positive and negative.

¹ <http://www.360buy.com/>

4.3 Experiment Setting

We have compared three groups of experiments to verify the effectiveness of our model. They are Baseline, Other Models, and Our Model.

- **Baseline:** After using BOW to represent the sentence, we treat Support Vector Machine (SVM) and Logistic regression (LR) models as baseline.
- **Other Models**
 - ✧ SNSS (Single Nucleus Single Satellite Method), following the idea of [2,11,13]. we get clause-level weights according to different clauses in contrast relation.
 - ✧ JS (joint strategy), the joint strategy used in [5]. Specifically, the joint strategy uses different BOW model to represent the different clauses of contrast sentences and learns the different BOW weights together.
- **Our Model:** we get the results of TLLR model. For the word-level weights, we first utilize the parameters of LR to initialize them, and then train weights according to Equation (9-10). The weights in clause level are set according to different parts in a contrast sentence with different connectives, and we use Equation (11) to get the parameters.

4.4 Experiment Results

The 5-fold cross-validation results are given in Table 3.

Baseline Group: We use the results of LR as the results of the comparative experiment in baseline system since our model is based on LR.

Other Models Group: Compared with the performance of LR baseline system (86.05%), the result of SNSS (86.00%) is decreased slightly. The reason is due to the drawbacks of the traditional pipeline method. As SNSS uses two-layer model which fixes word weights (from LR baseline) and just learns clause-level weights. The word weights in LR baseline are learned from contrast sentence dataset, which can be suitable for sentiment classification of contrast sentences. However, the word weights are probably not be suitable for clause sentiment classification. Therefore, the results of other models are slightly worse than the baseline. However, the result of JS is better than the baseline result, which shows different BOW model for different parts is useful for sentiment classification.

Table 3. Experiment results

Systems	5-Fold Cross Validation Results (%)					
	First	Second	Third	Fourth	Fifth	Average
SVM	86.00	86.00	83.00	85.00	85.50	85.10
LR	86.75	86.75	83.00	87.00	86.75	86.05
SNSS	86.25	88.00	83.50	86.00	86.25	86.00
JS	87.50	85.75	85.25	85.75	86.25	86.10
Our Model	86.58	87.70	84.95	86.73	87.73	86.74

Our Model Group: Compared with other models and the baseline system, our model gets better results by achieving an improvement by overall 0.7 point in the dataset. The reason our model is better than JS is that, JS is a simple version of TLLR. When we use one to fix the clause-level weights and just learn word-level weights, the TLLR model is equal to JS.

4.5 Parameter Analysis

In this section, the reasons why our model can work well will be presented. We set parameters in TLLR using the following methods. Since our model is a two-layer structure, we have to set the method to train word-level weights and clause-level weights separately.

For the word-level weights, we have two kinds of setting method. One is “Fix”, which fixes word-level weights through the parameters of LR baseline. The other is “Diff”, which first utilizes the results of LR to initialize them, and then trains the weights according to Equations (9) and (10). Compared with the two kinds of methods, we will confirm whether our joint model is better than the previous pipeline methods.

For the clause-level weights, we also have two kinds of setting methods. One is “Relation”, which sets clause-level weights according to the contrast relation. The other is “Connective”, which sets clause-level weights based on the different connectives. Compared with these two kinds of methods, we will find out whether the connectives are better than the contrast relation to capture which clause is more important for sentiment classification of contrast sentence.

- ✧ **FixRelation:** We fix our word-level weights through the parameters of LR baseline. We set the clause-level weights according to the different clauses in contrast sentence, and we use Equation (11) to get the parameters.
- ✧ **FixConnective:** We fix word-level weights through the parameters of LR baseline. The weights in clause-level are set according to the different clauses in contrast sentence for different connectives, and Equation (11) is used to get the parameters.
- ✧ **DiffRelation:** For the weights in word level, we first utilize the results of LR to initialize them, and then train the weights according to Equations (9) and (10). For the weights in clause level, the setting method is as the same as that in **FixRelation**.
- ✧ **DiffConnective:** For the weights in word level, the setting method is as the same as that in the **DiffRelation**. For the weights in clause level, the setting method is as the same as that in **FixConnective**.

Table 4 gives the results of the four methods in TLLR.

Table 4. Different parameter results in TLLR

Methods	5-Fold Cross Validation Results (%)					
	First	Second	Third	Fourth	Fifth	Average
FixRelation	86.25	88.00	83.50	86.00	86.25	86.00
FixConnective	86.25	87.75	84.25	86.75	87.25	86.45
DiffRelation	87.16	87.80	84.16	86.33	86.56	86.40
DiffConnective	86.58	87.70	84.95	86.73	87.73	86.74

Learning all Weights vs. Learning Clause-Level Weights: From Table 4, we can find the result of DiffRelation (86.40%) is better than the result of FixRelation (86.00%). The result of DiffConnective (86.74%) is also better than the result of FixConnective (86.45%). Based on the previous comparisons, we can find that the model, which learns word-level weights and clause-level weights jointly, is better than the model that only learns clause-level weights for the task of contrast sentence sentiment classification.

Connective vs. Relation: Which is the best parameter to model the clause sentiment label into sentence sentiment label, the connective or the relation? From Table 4, we can find the average results of FixConnective (86.45%) are better than the average results of FixRelation (86.00%), and the average results of DiffConnective (86.74%) are better than the average results of DiffRelation (86.40%). Therefore, we find the answer for that question is the connective, not the relation. Compared with the relation, connectives can be more correct and careful to reflect which part is more important for the contrast sentences. To analyze the problem more carefully, we get the average clause-level weights of 5-fold cross-validation in Table 5.

Table 5. Clause-level weights in TLLR Model

Type Model		Satellite	Nuclei
Fix Relation	转折 Contrast	0.48	0.52
Fix Connective	不过 however	0.6	0.4
	但 but	0.46	0.54
	但是 but	0.41	0.59
	只是 yet	0.56	0.44
	可是 however	0.41	0.59
Diff Relation	转折 Contrast	0.45	0.55
Diff Connective	不过 however	0.58	0.42
	但 but	0.42	0.58
	但是 but	0.36	0.64
	只是 yet	0.55	0.45
	可是 however	0.34	0.66

For the model we fix word level weights (FixRelation and FixConnective), we can find that our model can capture that the nuclei clause (0.52) in sentence is more important than the satellite clause (0.48) for sentiment classification in the contrast relation. For the connectives, our model can capture more information. For connectives “但是|but” and “但|but”, the nuclei clause (0.67, 0.64) is more important than the satellite clause (0.33, 0.36), however, for connectives “只是|yet” and “不过|however”, the results are different. The satellite clause (0.53, 0.80) is more important than the nuclei clause (0.47, 0.20). The conclusion is also consistent with the situation

when we use these connectives. We can also get the same conclusion from Table 5 for DiffRelation and DiffConnective.

5 Discussion and Analysis

Some case studies are discussed to illustrate the advantages of our model in this section, which are shown in Figure 4 and Figure 5.

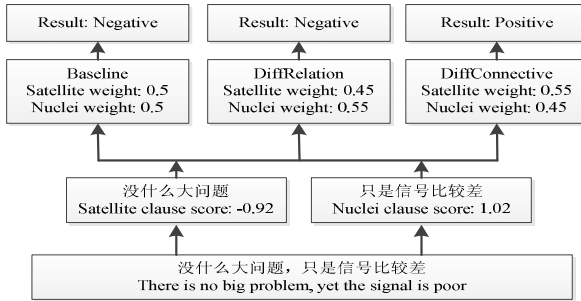


Fig. 4. An example about ‘只是lyet’

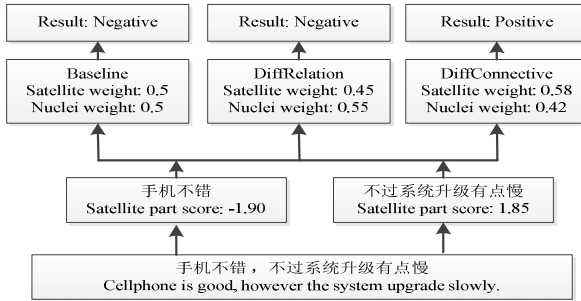


Fig. 5. An example about ‘不过however’

The first example in Figure 4 is about the connective ‘只是lyet’. The standard sentiment label for the example is positive. Different scores result in different polarities. When the satellite clause score is bigger than 0, then the polarity of the clause is positive, and vice versa. Baseline parameter will set the satellite and nuclei weight to 0.5, then the sentence score will bigger than 0, therefore the result for Baseline is negative. DiffRelation parameter will also get the same result. However, when we set clause weight using DiffConnective parameter, the result will be reverse. For the connective ‘只是lyet’, the satellite clause is important than the nuclei clause in DiffConnective parameter, which makes the method get the right answer.

The second example is about the connective ‘不过however’. The standard sentiment label for the example is positive. We can get the similar result of analysis from Figure 5.

From the previous analysis, we can find the following conclusion. When the connective in a contrast sentence is ‘只是|yet’ or ‘不过|however’, our model can work better than related work. From the connective distribution in contrast relation in Table 3, the percentage of the two connectives are about 43%. Therefore, it is useful to consider different clause-level weights with different connectives.

6 Conclusion and Future Work

To address sentiment classification of contrast sentence, this paper proposes a TLLR model. Compared with the existing pipeline methods, TLLR model is a global model, which learns different level weights jointly. For the clause-level weights, TLLR model is more careful to capture the importance of the nuclei and satellite clause in the sentence for the contrast relation. For the word-level weights, TLLR model can learn more adequate weights for the clause-level sentiment label. From the experiment results, we find that our model is better than the baseline and all other existing models. Our contributions can be summarized as follows:

- (1) TLLR model considers different clause-level weights with different connectives, not just relations.
- (2) TLLR model can learn word weights and clause-level weights together, which can avoid the drawbacks in the pipeline method.
- (3) Our experiment results confirm different connectives have different influence on overall sentiment classification.

The future work can be considered in two lines. The first line is to attempt to help sentiment classification by using other relations. The second line is to incorporate aspect into sentiment classification of contrast sentence. How to deduce the overall polarity from the polarity of each aspect is need to do for the future work.

Acknowledgements. We thank the three anonymous reviewers for their helpful comments and suggestions. We thank Haitong Yang for his help with model construction. We thank Chengqing Zong for his grateful comments on an earlier draft. The research work has been funded by the High New Technology Research and Development Program of Xinjiang Uyghur Autonomous Region under Grant No.201312103.

References

1. Balahur, A., Mihalcea, R., Montoyo, A.: Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications. *Computer Speech & Language* 28(1), 1–6 (2014)
2. Heerschop, B., Goossen, F., Hogenboom, A., Frasinca, F., Kaymak, U., de Jong, F.: Polarity analysis of texts using discourse structure. In: *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1061–1070 (2011)

3. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22(2), 110–125 (2006)
4. Kim, S.-M., Hovy, E.: Determining the Sentiment of Opinions. In: *Proceeding of the International Conference of Computational Linguistics (COLING)*, pp. 1367–1373 (2004)
5. Li, S., Huang, C.-R.: Sentiment Classification Considering Negation and Contrast Transition. In: *Proceedings of the Pacific Asia Conference on Language, Information, and Computation, PACLIC (2009)*
6. Liu, B.: Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies* 5(1), 1–167 (2012)
7. Narayanan, R., Liu, B., Choudhary, A.: Sentiment Analysis of Conditional Sentences. In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86 (2009)
8. Pang, B., Lee, L., Vaithyanatha, S.: Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86 (2002)
9. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 271–278 (2004)
10. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval* 2(2), 1–135 (2008)
11. Taboada, M., Voll, K., Brooke, J.: Extracting sentiment as a function of discourse structure and topicality. *Simon Fraser University, Tech. Rep.*, 20 (2008)
12. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 417–424 (2002)
13. Wang, F., Wu, Y., Qiu, L.: Exploiting Discourse Relations for Sentiment Analysis. In: *Proceeding of the International Conference of Computational Linguistics (COLING)*, pp. 1311–1319 (2012)
14. Xia, R., Hu, X., Lu, J., Yang, J., Zong, C.: Instance Selection and Instance Weighting for Cross-Domain Sentiment Classification via PU Learning. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2176–2182 (2013)

Emotion Classification of Chinese Microblog Text via Fusion of BoW and eVector Feature Representations

Chengxin Li, Huimin Wu, and Qin Jin*

School of Information, Renmin University of China, Beijing, China, 100872
{2011202429, whmliu, qjin}@ruc.edu.cn

Abstract. Sentiment Analysis has been a hot research topic in recent years. Emotion classification is more detailed sentiment analysis which cares about more than the polarity of sentiment. In this paper, we present our system of emotion analysis for the Sina Weibo texts on both the document and sentence level, which detects whether a text is sentimental and further decides which emotion classes it conveys. The emotions of focus are seven basic emotion classes: anger, disgust, fear, happiness, like, sadness and surprise. Our baseline system uses supervised machine learning classifier (support vector machine, SVM) based on bag-of-words (BoW) features. In a contrast system, we propose a novel approach to construct an emotion lexicon and to generate a new feature representation of text which is named emotion vector eVector. Our experimental results show that both systems can classify emotion significantly better than random guess. Fusion of both systems obtains additional gain which indicates that they capture certain complementary information.

Keywords: Emotion Classification, Sentiment Analysis, Sentiment lexicon, Text Feature Representation.

1 Introduction

Research of sentiment analysis of microblog texts has shown great research value, owing to its comprehensive applications in many fields, from earlier work by Pang about sentiment analysis of movie reviews [1] to nowadays more and more important applications in other fields such as business decision [2], politic election [3, 4] etc.

Weibo, short for Microblog in Chinese, has several aspects that are different from the traditional long texts such as movie reviews in sentiment analysis. Firstly, it is short with no more than 140 Chinese words. Because its shortness, it has been regarded as a convenient tool to use and to share daily life thus produce a large quantity of data for research. However, shortness also makes it harder for sentiment analysis compared with long texts. Secondly, Chinese is mainly used in Weibo instead of English. Chinese is largely different from English to some degree, like the character or the sentence structure, so the sentiment analysis work done with English microblogs like twitter may not be directly applied to Chinese microblog analysis. Thirdly, multi-

* Corresponding author.

sentiments in Weibo are more confusable. In formal long texts, which are regulated by conventional article rules, though multi-sentiment might also exist, we can determine the major emotion and the minor emotion by article rules, which is not effective in Weibo texts analysis. Fourthly, words used in Weibo are more casual than in long formal texts. For examples, there are web popular words like “麻麻”, “跪了” or emotion expressions like 😊 (corresponding input “[笑]”) and 😨 (corresponding input “[惊恐]”) in Weibo texts. Web popular words might be shown in traditional characters but with different meanings or emotions. For example, “跪了” refers to an action with no emotion polarity traditionally, but now it equals to a frustrating emotion. Moreover, some of these web popular words have several different meanings and different emotions owing to their informality. Emotion expressions are provided to make the Weibo texts more interesting. They are inputted by input tools like Sogou input, Baidu input, etc. How to effectively deal with free web texts is very important for Weibo sentiment analysis.

Due to its popularity of Weibo usage in China society, sentiment analysis of Weibo texts are becoming more and more important. The conference on Natural Language Processing and Chinese Computing (NLPCC) 2014 holds several evaluation tasks in natural language processing and Chinese computing. “Emotion Analysis in Chinese Weibo Texts” is one of the evaluation tasks. This paper presents our work of emotion classification on Weibo texts.

The rest of this paper is organized as follows. Section 2 introduces some related work. Section 3 describes the two systems we build and section 4 presents the experimental results. Section 5 presents some conclusions and future work.

2 Related Work

We can broadly summarize the previous research in sentiment analysis into three categories: analysis based on rules, analysis based on unsupervised classification and analysis based on supervised classification. Of the three categories, the last one performs relatively better.

Rule based analysis is mainly performed together with the emotion lexicon. In English texts analysis, Kamps used synonyms in WordNet lexicon [5]. In Chinese texts analysis, Yanlan Zhu used the HowNet Chinese lexicon [6]. Chun Li also used HowNet but construct another list consisting of the seed words which are further used to get the polarity of words in document [7]. In addition to the emotion lexicon, the researchers also noticed the influence of grammar, like the negative words, the adversatives and degree adverbs. They combine some of these words in several lists determined by their function and add these effects to the analysis process. The performance of this method is largely decided by the size and applicability of the emotion lexicon.

Analysis based on unsupervised classification is proposed by Turney [8] with some templates to extract the adjectives and adverbs as the emotion phrase, which further determine the PMI (Pointwise mutual information) and SO (Semantic orientation). The method deeply depends on the template of the basic words, thus has great limitations.

Analysis based on supervised classification gets the emotion classification model using labeled training data. The trained model is used to predict the emotion category of the test data. It is first proposed by Pang and Lee in 2002 [1]. The baseline algorithm adapted from it usually contains three modules: Tokenization, feature extraction and classification. The classification uses different classifiers like Naïve Bayes [9], Maximum Entropy (MaxEnt) [10] and Support Vector Machine (SVM) [11].

In this paper, we build our baseline system with supervised machine learning classifier SVM based on bag-of-words (BoW) features. In our contrast system, we propose a new method to construct an emotion lexicon and then to generate a new feature representation based on the emotion lexicon. The two systems are combined for better emotion classification performance.

3 System Description

We build two systems for emotion classification of Weibo texts, one uses supervised learning approach based on traditional bag-of-words feature representation and the other also uses supervised learning approach but based on a new feature representation via emotion lexicon. We fuse two systems via late fusion on the classification score.

3.1 Baseline System Based on Bag-of-Word Feature Representation

The baseline system uses supervised learning approach support vector machine (SVM) based on bag-of-words (BoW) feature representation. There are two main phases in the emotion analysis process. The first phase is to detect whether a Weibo document is sentimental/emotional. The second phase is to classify the document into its proper emotion category if it is sentimental. Both the two phases consist of tokenization, feature extraction and supervised classification steps.

Tokenization: We use all words in a document for analysis, not only adjectives [12], but also verbs, adverbs, nouns, etc. We use Jieba [13], a Chinese text segmentation module built for python programming, for word segmentation.

Feature Representation: BoW model is widely used in text processing applications. It processes texts without considering the word order, the semantic structure or the grammar. The vector representation of BoW is a normally used feature representation for text document. The vocabulary is commonly selected using TF-IDF theory.

In our experiment, the BoW feature representation can take different vector values. The first kind of vector representation consists of only 0 or 1 value for each dimension, where 1 stands for the occurrence of a vocabulary word and 0 stands for non-occurrence. The second kind of vector representation consists of a real number for each dimension, where each real number stands for the frequency of one vocabulary word in a document.

Besides the top frequent words selected based on TF-IDF, we also consider the emotion expressions like “[笑]”, “[惊恐]” in the weibo texts and use them as another

vocabulary for feature representation. Emotion expressions are provided to make the weibo texts more interesting. These emotion expressions form a new vocabulary list from which we can create a new BoW feature representation. Emotion expression can be a very useful cue for sentiment analysis. We observe on the training data that 98% of the documents with emotion expressions are labeled with certain emotion classes. Moreover, repeat usage of punctuation like exclamatory mark or question mark can also be efficient cues for emotion detection. In our baseline system, we combine the word vocabulary selected based on TF-IDF, the vocabulary of emotion expressions and the vocabulary of punctuation marks to create the BoW feature representation. We assign different weights when combining the words vocabulary and the emotion expression plus punctuation marks vocabulary. The weights are tuned on held out development data.

In experiments shown in this paper, for emotion detection, we select top 500 most frequent words based on TF-IDF from both the documents with none emotion and the documents with emotions respectively. That leads to a vocabulary with 1000 words. After deleting repeated words, we get 610 words in the vocabulary. For emotion classification, we get the top 100 most frequent words from each of the seven emotion classes and this leads to a vocabulary of 700 words. After deleting repeated words, we get a word vocabulary of 560 words. This word vocabulary is combined with the emotion expression vocabulary containing 469 expressions plus the punctuation vocabulary with 7 punctuation marks for generating the BoW features.

Classifier Trained with Supervised Approach: We use Support Vector Machine (SVM) as our classifier. In our experiments, we use the LIBSVM Toolkit [14]. We tune related parameters through cross validation.

3.2 Contrast System Based on Emotion Vector Feature Representation

We construct a new emotion lexicon in this contrast system. We observe that in Weibo texts, we can roughly categorize different words into three types. Taking the sentences with “anger” emotion as example, the first type of words are “emotional words” like “怒” (angry) and “高兴” (happy) which are typical frequent words for expressing certain emotion. The second type of words are “common words” like “真的” (really) which commonly appear in documents but do not usually contain emotion inclination. The third type of words are “Not Emotional and Uncommon words” like “资本家” (capitalist). Based on our intuition, we expect the three types of words may have the following distribution as shown in Table 1, where n_i refers to the occurrence number of a word in documents of certain emotion class (for example, the emotion “anger”), n_o refers to the occurrence number of this word appears in other emotions (for example, emotions except “anger”), n_l refers to the number of emotion classes that this word appears in (for example, if this word only appears in documents of “anger” class, then n_l is 1. If it appears in documents of “like” and “happiness” classes, then n_l is 2).

We then use the following formula to compute the weight of every word and rank them in descending order:

$$weight = \frac{n_i}{(n_o * n_l + 1)} \quad (1)$$

We expect “Emotional words” should be ranked in the top, “Common words” should be ranked in the bottom, and “not Emotional but Uncommon words” should be ranked in the middle. The result proves that our intuition is relatively correct. Some examples for the emotion class “anger” are shown in Table 2.

Table 1. Expected distribution pattern of three types of words

<i>Type</i>	<i>n_i</i>	<i>n_o</i>	<i>n_l</i>
Emotional words	more	less	Less
Common words	fair	more	More
Not emotional but uncommon words	less	less	Less

Table 2. Word examples in ranked list for anger class

<i>Anger</i>	<i>Word (translation)</i>	<i>weight</i>
Examples in the top part of the ranked list	恨死 (hate)	12.3
	气死我了 (piss me off)	8.0
	MB (fuck)	7.0
	贱人 (bitch)	5.0
	这蛋 (bullshit)	5.0
Examples in the middle part of the ranked list	心肝儿 (darling)	0.5
	掩护 (cover)	0.5
	私车 (personal car)	0.5
	扭转 (reverse)	0.5
	秒钟 (clock)	0.5
Examples in the bottom part of the ranked list	挺 (very)	0.0031
	每 (every)	0.0029
	现场 (on site)	0.0029
	滴 (a drop)	0.0029
	害羞 (shy)	0.0029

In creating the emotion lexicon using above method, we don't include emotion expressions. After building the emotion lexicon, we use emotion expressions for generating feature representation like BoW. Inspired by the work in [15], which represents emotion words by a vector and every dimension of the vector represents a kind of emotion, if a word relates to the emotion to some extent, the corresponding dimension is 1, otherwise is 0. Similarly, we represent each text by an emotion vector (eVector) composed of 7 dimensions instead of hundreds of dimensions. The vector is in the format as follows:

$$eVector = (d_1, d_2, d_3, d_4, d_5, d_6, d_7) \quad (2)$$

where the seven dimensions correspond to the seven emotions of anger, disgust, fear, happiness, like, sadness and surprise respectively. The value of d_i is sum of all the words' weights (computed as in formula (1)) according to the emotion lexicon for each emotion class i (for example, anger is in 1st emotion class, disgust is in 2nd emotion class, etc).

4 Experiment and Analysis

4.1 Data Description

The data in this paper is collected from Sina Weibo (a popular Chinese Microblog site). The text of Microblog is labeled "none" if it does not convey any emotion. If the text conveys emotion, it is labeled with emotion categories from anger, disgust, fear, happiness, like, sadness, or surprise. The text is labeled with major emotion and minor emotion. Every Microblog document includes at least one sentence. Every sentence in a Microblog document is also labeled with corresponding emotions (major and minor). It is not necessary that every sentence conveys emotion in an emotional document. Therefore, it is possible that sentences in an emotional document can have "none" labels. The number of documents and the number of sentences for each emotion category in training data and test data are described in Table 3 and 4. We can see from the tables that the distribution of different emotion classes is not balanced.

We extract 469 emotion expressions from the training data and some examples are shown in Table 5. The value means the number of occurrence of certain emotion expression in documents with different emotion class labels. Admittedly, there is some informal usage, for example [抓狂] appears in both sentimental and non-sentimental documents. However, because the number of this emotion expression appears in sentimental category far more than in non-sentimental category, we think this emotion expression is still useful in emotion detection, though it may not be so effective in emotion classification.















Table 3. Number of documents for each emotion type in training and test data

<i>emotion type</i>	<i>training data</i>		<i>test data</i>	
	number	percentage	number	percentage
none	6591	47.0%	3603	60.1%
anger	669	4.8%	128	2.1%
disgust	1392	10.0%	389	6.5%
fear	148	1.1%	46	0.8%
happiness	1460	10.4%	441	7.3%
like	2204	15.7%	1042	17.3%
sadness	1174	8.4%	189	3.2%
surprise	362	2.6%	162	2.7%

Table 4. Number of sentences for each emotion type in training and test data

emotion type	training data		test data	
	number	percentage	number	percentage
none	29731	65.4%	11871	75.1%
anger	1899	4.2%	244	1.6%
disgust	3130	6.9%	679	4.3%
fear	299	0.7%	67	0.4%
happiness	2805	6.2%	641	4.1%
like	4259	9.4%	1630	10.4%
sadness	2478	5.4%	302	1.9%
surprise	820	1.8%	259	1.7%

Table 5. Number of occurrence of emotion expressions across difference emotion classes

Icon	input	none	Ang	Dis	Fea	Ha p	Lik	Sad	Sur	main
	[抓狂]	5	24	24	5	14	18	37	0	disgust
	[耶]	8	1	0	0	30	13	1	1	happy
	[鼓掌]	6	1	4	0	29	29	2	0	happy
	[委屈]	0	2	0	1	2	5	14	0	sad
	[泪]	14	15	16	6	31	22	156	4	sad
	[爱你]	5	0	3	0	33	42	3	0	happy
	[good]	6	0	1	0	9	24	2	5	like
	[吃惊]	1	2	2	1	1	1	3	14	surprise
	[偷笑]	18	1	8	1	63	34	3	1	happy
	[吐]	1	3	6	0	0	1	2	1	disgust
	[哈哈]	21	0	9	0	121	26	2	3	happy
	[心]	23	1	0	0	37	45	10	1	like
	[酷]	7	0	3	1	15	12	2	4	happy
	[眼泪]	0	0	0	0	1	1	4	0	sad

4.2 Evaluation Metrics

The evaluation metrics used in this paper are precision, recall and F-measure for emotion detection and looseAP and strictAP for emotion classification. In the evaluation, the system is required to produce the emotion classification results for both major emotion

and minor emotion. In our experiments, an emotion of an emotional document is decided by the comparison of scores for all seven emotion classes. The top ranked emotion class is the major emotion classification decision and the second top ranked emotion class is the candidate for minor emotion classification decision. If the difference between the major and minor emotion is larger than certain threshold, the minor emotion will be “none” instead of the second top ranked emotion class. The threshold can be tuned with cross validation, which is 0.5 in the experiments in this paper.

looseAP and strictAP are the two metrics used in the evaluation. On loose metric, the system will get a score of 1 if it correctly identify either the top emotion class or the minor emotion class. As for the strict metric, for the case that the ground truth contains both major and minor emotions, a system will get score 1 if both the major and minor emotion classes are matched. It will get a score of 0.666 if the major emotion class is identified and score 0.333 if the minor emotion class is identified. If the major emotion class hypothesis matches the minor emotion class in ground truth, it will get a score of 0.333 as well.

4.3 Experimental Results

As we have described in section 3.1, we use two BoW features in the baseline system: occurrence vs. frequency. The results based on these two types of BoW features are compared in Table 6 on both the document level (D) and sentence level (S). We can see from the results that frequency BoW is not largely different from occurrence BoW in terms of emotion classification performance. We expect the reason is that the document is too short, most words occur in the document only once, so frequency is either 0 or 1, thus the two feature representations are very close to each other.

Table 6. Baseline system performance with Occurrence vs Frequency BoW

	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>	<i>looseAP</i>	<i>strictAP</i>
<i>Occurrence (D)</i>	0.58	0.73	0.65	0.44	0.40
<i>Frequency (D)</i>	0.58	0.74	0.65	0.43	0.39
<i>Occurrence (S)</i>	0.58	0.50	0.54	0.33	0.31
<i>Frequency (S)</i>	0.57	0.52	0.56	0.34	0.32

Table 7. Confusion Matrix of baseline system on document level with occurrence BoW

	<i>none</i>	<i>anger</i>	<i>disgust</i>	<i>fear</i>	<i>happy</i>	<i>like</i>	<i>sad</i>	<i>surprise</i>
<i>none</i>	2363	15	176	0	142	796	89	22
<i>anger</i>	38	38	16	0	2	28	3	3
<i>disgust</i>	170	16	69	0	15	92	19	8
<i>fear</i>	15	1	8	0	2	14	5	1
<i>happy</i>	63	0	7	0	271	87	9	4
<i>like</i>	274	9	25	0	126	570	31	7
<i>sad</i>	54	3	17	0	7	33	72	7
<i>surprise</i>	40	5	23	0	10	35	3	46

Table 7 shows the confusion matrix of the baseline system on the document level with occurrence BoW (column is the ground truth and row is the system decision). From table 7, we can see that “none” is on average the most confusable class to all emotion classes, which indicates that our emotion detection step at the first place should be improved. We can also see that for some emotion classes, their confusable emotion classes are intuitively related classes, for example, “like” and “happy” are confusable. “surprise” is confusable with “like” or “disgust” depending on it is a good or bad surprise.

Table 8 presents the emotion classification results of the baseline system (based on occurrence BoW), the contrast system (based on eVector feature representation) and the fused system. The results show that fusion of both systems improves the emotion classification performance on both the document level and sentence level. As shown in previous section in Table 3 and 4, the distribution of emotion classes is not balanced. In Table 9, we therefore also compute the looseAP and strictAP performance with weights proportional to the number of documents/sentences in a particular emotion class.

Table 8. System performance of emotion classification on both document and sentence level

<i>System</i>	<i>Document Level</i>		<i>Sentence Level</i>	
	<i>looseAP</i>	<i>strictAP</i>	<i>looseAP</i>	<i>strictAP</i>
<i>Baseline system (BoW)</i>	0.44	0.40	0.33	0.31
<i>Contrast system (eVector)</i>	0.38	0.34	0.28	0.27
<i>Fusion</i>	0.46	0.41	0.34	0.32

Table 9. System performance of emotion classification on both document and sentence level with weighted AP computation

<i>System</i>	<i>Document Level</i>		<i>Sentence Level</i>	
	<i>looseAP</i>	<i>strictAP</i>	<i>looseAP</i>	<i>strictAP</i>
<i>Baseline system (BoW)</i>	0.65	0.59	0.69	0.65
<i>Contrast system (eVector)</i>	0.58	0.51	0.65	0.61
<i>Fusion</i>	0.66	0.60	0.72	0.68

We notice that some emotion categories are closely related. In many cases “anger” may express certain level of “disgust”, “like” may express certain level of “happiness”. Therefore, when the system classifies some “anger” as “disgust”, or “like” as “happiness”, we should not simply say it is wrong. However, if “anger” is recognized as “like” or “happiness”, there is no question that it is wrong, because they are totally opposite emotion categories. We therefore also look at the performance if we tolerate “anger” and “disgust” to belong to the same category, “happiness” and “like” to belong to the same category for the baseline system, contrast system, and fused system as shown in Table 10. The performance is obviously improved with the tolerance for all systems and fusion improves performance.

Table 10. System performance of emotion classification on three emotion class

<i>System</i>	<i>Disgust+Anger</i>	<i>Happy+Like</i>	<i>Sadness</i>
<i>Baseline system(Bow)</i>	0.22	0.71	0.38
<i>Contrast system(eVector)</i>	0.26	0.69	0.28
<i>Fusion</i>	0.27	0.73	0.42

Table 11. Emotion detection on document level with different weighting over words vs expression plus punctuation vocabulary

<i>Weights (words/expression+)</i>	<i>correct</i>	<i>proposed</i>	<i>gold</i>	<i>precision</i>
0.1/0.9	1736	2990	2397	0.580602
0.2/0.4	1754	3010	2397	0.582724
0.3/0.7	1761	3032	2397	0.580805
0.4/0.6	1762	3024	2397	0.582672
0.5/0.5	1767	3035	2397	0.582208
0.6/0.4	1751	3019	2397	0.579993
0.7/0.3	1750	3022	2397	0.579087
0.8/0.2	1751	3020	2397	0.579801
0.9/0.1	1729	3002	2397	0.575949
1.0/0.0	1666	3002	2397	0.554963

As described in previous section 3.1, we combine the words vocabulary and emotion expression plus punctuation marks vocabulary for baseline BoW feature representation generated with different combination weights. Table 11 compares the emotion detection results with different combination weights. *Gold* refers to the total number of ground truth emotional documents. *Proposed* refers to the total number of system hypothesized emotional documents. *Correct* refers to the total number of correct system hypothesized emotional documents. We can see that weights ratio of 0.4/0.6 achieves best detection result. Please notice that the last row (weights 1.0/0.0) refers to the case that expression and punctuation marks are not used for BoW generation. Its worst performance proves that expression and punctuation are important cues for emotion analysis.

We also combine the baseline system which uses bag-of-word feature representation and the contrast system which uses eVector feature representation. Table 12 shows the different fusion weights for emotion classification on the document level. As the results show that fusion of the two systems with appropriate fusion weights achieves additional gain.

Table 12. Fusion weights of baseline and contrast systems on document level

<i>Weights (eVector/BoW)</i>	<i>looseAP</i>	<i>strictAP</i>
0.0/1.0	0.445	0.396
0.1/0.9	0.447	0.397
0.2/0.8	0.455	0.404
0.3/0.7	0.457	0.406
0.4/0.6	0.450	0.398
0.5/0.5	0.445	0.395
0.6/0.4	0.431	0.382
0.7/0.3	0.423	0.376
0.8/0.2	0.410	0.361
0.9/0.1	0.398	0.351
1.0/0.0	0.383	0.337

5 Conclusion

Weibo text which is the most popular social media in China has attracted much research interest in recent years. Emotion analysis which not only cares about the polarity of sentiment but also the detailed emotion category is a more challenging task. In this paper, we present our two systems for emotion analysis of Chinese Weibo texts on both the document and sentence level. The baseline system uses SVM as classifier based on bag-of-words features representation. The vocabulary for BoW generation combines words, emotion expression and punctuation marks. Experimental results confirm that emotion expression and punctuation marks are important cues for emotion analysis of Weibo texts. The contrast system proposes a new method to construct an emotion lexicon to generate a new feature representation, the emotion vector eVector. Our experimental results show that both systems can classify emotion significantly better than random guess. Fusion of both systems obtains additional gain, which indicates that they capture certain complementary information. In the future work, we will explore new methods to improve the emotion detection performance, enhance the proposed eVector feature representation by utilizing the emotion expressions as well. We will also investigate different classification approaches.

Acknowledgements. This work is supported by the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 14XNLQ01), the Beijing Natural Science Foundation (No. 4142029).

References

1. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, vol. 271. Association for Computational Linguistics (2004)

2. Li, N., Wu, D.D.: Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems* 48(2), 354–368 (2010)
3. Tumasjan, A., Sprenger, T.O., Sandner, P.G., et al.: Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In: ICWSM, vol. 10, pp. 178–185 (2010)
4. O'Connor, B., Balasubramanyan, R., Routledge, B.R., et al.: From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM* 11, 122–129 (2010)
5. Kamps, J., Marx, M.J., Mokken, R.J., et al.: Using wordnet to measure semantic orientations of adjectives (2004)
6. Zhu, Y.L., Min, J., Zhou, Y., et al.: Semantic orientation computing based on HowNet. *Journal of Chinese Information Processing* 20(1), 14–20 (2006); 朱嫣岚, 闵锦, 周雅倩, 等.: 基于HowNet的词汇语义倾向计算. *中文信息学报* 20(1), 14–20 (2006)
7. Dun, L., Fuyuan, C., Yuanda, C., et al.: Text Sentiment Classification Based on Phrase Patterns. *Computer Science* 35(4), 132–134 (2008); 李钝, 曹付元, 曹元大, 等.: 基于短语模式的文本情感分类研究. *计算机科学* 35(4), 132–134 (2008)
8. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics (2002)
9. Lewis, D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) *ECML 1998. LNCS*, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
10. König, R., Renner, R., Schaffner, C.: The operational meaning of min-and max-entropy. *IEEE Transactions on Information Theory* 55(9), 4337–4347 (2009)
11. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* 20(3), 273–297 (1995)
12. Benamara, F., Cesarano, C., Picariello, A., et al.: Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone. In: ICWSM (2007)
13. Lin, R.G., Tsai, T.C.: Scalable System for Textual Analysis of Stock Market Prediction. In: The Third International Conference on Data Analytics 2014, pp. 95–99 (2014)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3), 27 (2011)
15. Wu, Y., Kita, K., Ren, F., Matsumoto, K., Kang, X.: Exploring Emotional Words for Chinese Document Chief Emotion Analysis. In: Proceedings of PACLIC 2011, pp. 597–606 (2011)

Social Media as Sensor in Real World: Geolocate User with Microblog

Xueqin Sui¹, Zhumin Chen^{1,*}, Kai Wu¹, Pengjie Ren¹,
Jun Ma¹, and Fengyu Zhou²

¹ School of Computer Science and Technology,
Shandong University, Jinan, 250101, China

² School of Control Science and Engineering,
Shandong University, Jinan, 250002, China
chenzhumin@sdu.edu.cn

Abstract. People always exist in the two dimensional space, i.e. time and space, in the real world. How to detect users' locations automatically is significant for many location-based applications such as dietary recommendation and tourism planning. With the rapid development of social media such as Sina Weibo and Twitter, more and more people publish messages at any time which contain their real-time location information. This makes it possible to detect users' locations automatically by social media. In this paper, we propose a method to detect a user's city-level locations only based on his/her published posts in social media. Our approach considers two components: a Chinese location library and a model based on words distribution over locations. The former one is used to match whether there is a location name mentioned in the post. The latter one is utilized to mine the implied location information under the non-location words in the post. Furthermore, for a user's detected location sequence, we consider the transfer speed between two adjacent locations to smooth the sequence in context. Experiments on real dataset from Sina Weibo demonstrate that our approach can outperform baseline methods significantly in terms of *Precision*, *Recall* and *F1*.

Keywords: Location Detection, Social Media, Words Distribution over Locations.

1 Introduction

Location based services, such as dietary recommendation, shopping advertisement and travel routine plan, have increasingly become significant and popular not only for research but also for industry. How to detect users' regular locations automatically is necessary. New social media such as Twitter and Sina Weibo have spawned greatly as human-powered sensing networks. More and more users actively publish short messages about bits and pieces of their lives at any time and any places. This makes it possible to detect a given user's locations from

* Corresponding author.

his/her published posts in social media. Some people may argue that most posts in social media contain GPS information since most equipments used by users to publish posts, such as mobile phone, have GPS modules. However, there are only about 0.42% posts contain GPS information according to [1], because most users close their GPS modules. Thus, it is not easy to detect a user's locations automatically.

In this paper, we propose a method to detect a user's locations purely based on the content of his/her published Sina Weibo posts in the absence of any other geospatial cues. For a post of a given user, we combine two components, i.e. a Chinese location library and a Bayes model based on words distribution over locations, to judge whether it contains location information and further to detect the location. If the post contains an explicit location name of the location library, the first component is used to match it directly. If the post does not refer any location name, there maybe some words which can imply the location where the user is. Thus, the second component first learns a model of words distribution over locations from Wiki and use it to mine the implied location information under the non-location words in the post. Furthermore, for a user's detected location sequence, we compute the minimum transfer time between two adjacent locations and use it to smooth the sequence in context. Experiments on real dataset from Sina Weibo demonstrate that our approach can outperform baseline methods significantly in terms of *Precision*, *Recall* and *F1*.

The rest of this paper is organized as follows. We introduce related work in Section 2. Section 3 gives our method in detail. In section 4, we discuss the corresponding experiments. We make some conclusions in addition to introducing future works in Section 5.

2 Related Work

There's a large amount of previous work on location prediction. There are three kinds of works: Location prediction based on content, location prediction based on social relationship and application of predicted locations.

Location prediction based on content: In [1], it finds word geographical spatial distribution based on the method of probability. According to the results, the words are divided into the local words of position sensitive (local words) and the location is not sensitive to nonlocal term (non-local words). Then it is based on local words to find the location of the user. [2] studies the location identification problem in blog. First, for each post it uses a named entity identifier which is based on GeoName gazetteers to identify location in the entity. After that, it identifies the place name entity. At last, it uses an ontology to represent hierarchical relationships between the place names which are stored in GeoName gazetteer. [3] uses place name dictionary to identify the main places of a web page. First, it recognizes all mentioned place names in Web page. Then for each case it gives a place and the corresponding confidence level. Finally, the confidence level of the highest place will be as the main place of the whole page. [4] studies the spatial transformation problem in search engine query.

[5] builds a $m * n$ grid based on longitude/latitude coordinate. Each cell represents a grid position. It uses some pictures of location known as the training set in Flickr, according to the label whether in the GeoName place names library to determine whether the label on behalf of the specific geographical location, and training a probability model of the language based on the label marked by the user. For a given image, the method is to predict its geographic location based on this model. [6] puts forward a system combined with the use of text and visual features from 20 million images crawling from Flickr and then mapping to the map. In this paper, limiting 10 markers in a city, this method uses the 10 markers image as a positive example and others as a negative example to train a classifier, then uses the classifier to implement the classification of the image. [7] puts forward several supervised methods, only using text to mark a location for document. First, the earth's surface is divided into multiple cells according to the longitude and latitude. After that, it uses the training set to train term distribution, document distribution and cell distribution. Then [7] uses three supervised methods to choose one of the most possible cell for each document as the geographical location of the document.

Location prediction based on social relationship: [8] explores supervised and unsupervised learning scenarios method to predict location, concluding reconstructing the entire friendship graph with high accuracy even when no edges are given and inferring people's fine-grained location, even when they keep their data private and we can only access the location of their friends. Li et al. [9] proposes a system to integrate both network and content-based prediction via a unified discriminative influence model which combine locations that a Twitter user mentions with the locations of the user's followers. Backstrom et al. predicts the home address of Facebook users based on provided addresses of one's friends [10], Cho et al. focuses on modeling user location in social networks as a dynamic Gaussian mixture, a generative approach postulating that each check-in is induced from the vicinity of either a person's home, work, or is a result of social influence of one's friends [11]. [12] proposes a novel network-based approach for location prediction in social media that integrates evidence of the social tie strength between users for improved location prediction.

Application of predicted locations: [12] studies the application of location prediction in public health. [13] studies the application of location in earthquake and other disasters. From [12,13] you can say that location prediction is very important in our real world.

3 Content-Based Location Detection

We first give a general description of the location detection problem with Sina Weibo.

Definition. Given a post bo_i published by a Sina Weibo user u , Our purpose is to predict the probability $p(l_j|bo_i)$ that user u is at the location l_j . The location with maximum probability is predicted as the user's location $l_{est(i)}$ when he publishes the post.

$$l_{est(i)} = l_j = \arg \max_{l_j \in L} P(l_j|bo_i) \quad (1)$$

where L denotes all the locations in the predefined Chinese location library.

In this paper, $P(l_j|bo_i)$ is defined as:

$$P(l_j|bo_i) = \alpha \cdot P_g(l_j|bo_i) + (1 - \alpha) \cdot P_w(l_j|bo_i) \quad (2)$$

The function above consists of two components: $P_g(l_j|bo_i)$ and $P_w(l_j|bo_i)$ which are computed based on the location library and the Bayes model of words distribution over locations respectively. α is a predefined tradeoff parameter and its optimized value is determined by experiment. In this paper, we set $\alpha = 0.5$. Table 1 shows the result of $F1$ when α has different values.

Table 1. The values of $F1$ when α has different values

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$F1$	0.4685	0.4805	0.4734	0.4713	0.5863	0.5086	0.4688	0.4753	0.4780

A post may refer to multi locations. The $P_g(l_j|bo_i)$ is computed as:

$$P_g(l_j|bo_i) = \frac{f(l_j)}{\sum_{l_q \in L} f(l_q)} \quad (3)$$

where $f(l_j)$ represents the frequency of the location l_j mentioned in the post bo_i . $\sum_{l_q \in L} f(l_q)$ is the frequency of all locations in the Chinese location library L mentioned by bo_i .

A post may not contain any direct location names of the library. But there are some words which can imply the location where the user is. For example, when a user publishes a post ‘‘I am watching games at GuoAn home.’’ It is well known that ‘Beijing’ is the home of GuoAn team. Then, we can refer the location is ‘Beijing’ because the probability that ‘GuoAn home’ coexists with ‘Beijing’ is very high. Thus, $P_w(l_j|bo_i)$ can be computed with words distribution over locations:

$$P_w(l_j|bo_i) = \frac{P(bo_i|l_j) \cdot P(l_j)}{P(bo_i)} \quad (4)$$

where $P(bo_i)$ denotes the prior probability of the post, and is the same for any post. So,

$$P_w(l_j|bo_i) \propto P(bo_i|l_j) \cdot P(l_j) \quad (5)$$

where $P(bo_i|l_j)$ denotes the probability that the location l_j generates the post bo_i and $P(l_j)$ is the prior probability of the location l_j mentioned by users.

$P(bo_i|l_j)$ can be computed as:

$$P(bo_i|l_j) = \prod_{w_s \in BO_i} P(w_s|l_j) \quad (6)$$

where bo_i is segmented into word set BO_i with the tool ICTCLAS¹ and w_s is a word of the post bo_i . $P(w_s|l_j)$ denotes the the probability of words distribution over locations.

We observe that there are many words such as local snacks and local scenic spots can reveal where the user is. In order to get the probability of word distribution over locations, we collect data from the Chinese Geography in Wiki² to train a Bayes model to compute $P(w_s|l_j)$. The Geography contains all Chinese cities and there is a detailed description for each city such as administrative division, government name, traffic, school, history, hospital, tourist attraction and so on. Finally, we collect 372 city-level locations and corresponding 6355 words (denoted as W) with typical location character. $P(w_s|l_j)$ is calculated as:

$$P(w_s|l_j) = \frac{\log \text{count}(w_s)}{\sum \log \text{count}(w_h)}; \quad w_s, w_h \in \{W \cap BO_i\} \quad (7)$$

where $\text{count}()$ denotes the word frequency in bo_i .

For different locations, values of $P(l_j)$ are different because their city levels, ranges and population are different. We assume the popularity of a location can be evaluated by the number that it is indexed by a search engine. Therefore, $P(l_j)$ is computed as:

$$P(l_j) = \frac{\log \text{count}(l_j)}{\sum_{l_q \in L} \log \text{count}(l_q)} \quad (8)$$

where $\text{count}(l_j)$ represents the result number returned by Google when we submit l_j to Google. Intuitively, the higher the number is, the higher the popularity is.

There is a strong chronological order and context in a user's location sequence. Thus, we further consider the speed of transports to smooth the posts with abnormal locations and the posts without detected locations by using their adjacent locations. We observe that even by an airplane, it is impossible to transfer from one place to another faraway place within a limited time interval. Thus, we compute the threshold time of a person from one location to another as follows:

$$th(l_j, l_k) = \frac{Dis(l_j, l_k)}{v} \quad (9)$$

$th(l_j, l_k)$ denotes the time interval that a user moves from location l_j to location l_k . $Dis(l_j, l_k)$ is the distance in Kilometers between locations l_j and l_k . v is the speed of a kind of transport. Because it is difficult to decide which kind of transport the user takes, in this paper we consider the speed of airplane because it is the fastest transport and set $v = 800$ Kilometers. Thus, $th(l_j, l_k)$ is the minimum time that a user transfers from l_j to l_k .

For a user's two detected adjacent locations l_j and l_k , they correspond to two posts which have the property of publishing time t_j and t_k respectively. We can

¹ <http://ictclas.org/>

² <http://en.wikipedia.org/wiki/China#Geography>

get the moving time interval $t(l_j, l_k) = t_k - t_j$. If $t(l_j, l_k) < th(l_j, l_k)$, then we set $l_k = l_j$.

4 Experiments

4.1 Data Set

We have to contrast the dataset since there is no standard corpus for evaluating location prediction problem with social media. We collect real data from Sina Weibo. We first select 1000 active users and crawl all their posts published from Aug. 2009 to Apr. 2014. Then if the number of a user's published post is smaller than 10, we remove the user. After that, there remains 772 users. Finally, we annotate all posts with Chinese location library manually. The specific of the data set is shown in Table 2.

Table 2. Dataset from Sina Weibo

item	number
users	772
posts	826,018
locations	372
posts with annotated location	304,384

4.2 Baseline Methods

There are three baseline methods we compare with in this paper. The first method only uses the Chinese location library [2], represented as GL_{lib} . The second method is only based on the probability model [4], denoted as GL_{pm} . The third baseline method is the improved Probability model, represented as GL_{ipm} , which is extended from the second method with the transfer speed smoothing.

4.3 Evaluation Measures

We use standard measures *Precision*, *Recall* and *F1* to evaluate the user's geolocation results. If the detected location generated by the methods agrees with the manually annotated location, we view it as a correct geolocation.

Precision is the fraction of detected locations that are correct.

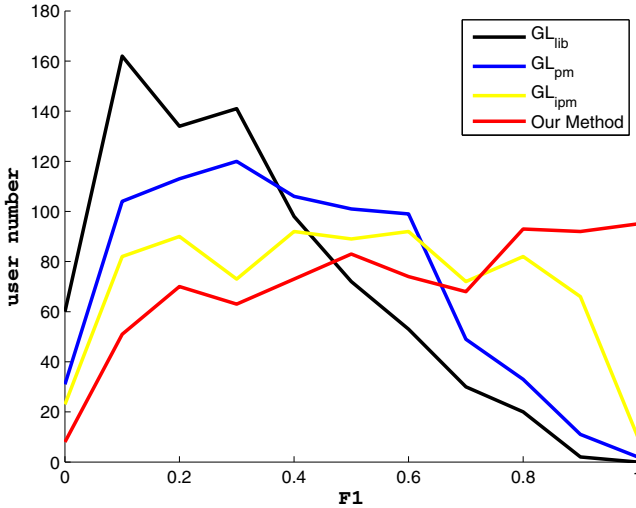
Recall is the fraction of correct locations that are detected.

F1-score is calculated using following function: $F1 = 2 * (Precision * Recall) / (Precision + Recall)$.

Note that all measures above are macro-average for all users.

Table 3. Experiment results

Methods	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
GL_{lib}	0.8226	0.2029	0.2989
GL_{pm}	0.8394	0.2649	0.3802
GL_{ipm}	0.8335	0.3681	0.4795
Our Method	0.8231	0.4991	0.5863

**Fig. 1.** Distribution of the number of users over $F1$

4.4 Results and Discussion

Table 3 shows the result of our method and the three baseline methods. It is obvious that our method achieves the best performance. Although the *Precision* of our method is slightly less effective than GL_{pm} and GL_{ipm} , the *Recall* and *F1* of our method are significantly outperform that of baselines. Our approach improves the *Recall* significantly, it performs about 1.5 times than the second method. It is because that our method considers the smoothing technology, including removing abnormal locations and adding some detected locations. However, the values of *Recall* for all methods are not very high. The reason is that we consider all posts of a user that we can get the actual locations, but a lot of posts don't contain any location information. Therefore, we can't predict locations of these posts.

We can also see that the *Recall* of GL_{ipm} is 6% higher than that of GL_{pm} . This further confirms the effectiveness of the smoothing technology. We further have a look at the posts we predict false. They have these features: (i) there are many location names mentioned in the post. (ii) the nuptial problem. For

example, ‘Chaoyang’ is not only a city in Liaoning province but also a county in Beijing.

To further evaluate micro-performance of the four methods, we analyse the location detection performance of each user. Fig.1 shows the distribution of the number of users over the measure $F1$. It is obvious that when $F1 > 0.7$, the curve of our method is above those of the other methods. This means that our method can achieve good performance for most users, for the reason that many posts contain explicit location names and many contain some information about locations. The GL_{lib} can mine explicit location names and GL_{pm} can mine implicit location names. However, our method can mine not only explicit location names but also implicit location name, therefore, our method can achieve a better performance than others.

5 Conclusion

In this paper, we have proposed an approach to detect users’ locations automatically using their published posts in social media. Our method considers both the direct matching with location name and the undirect mining of implied word distribution over locations. The transfer speed between locations is also utilized to smooth the detected location series. We implement both three baseline methods and our new method, and experimentally verify that our method can outperform the baselines especially in terms of the measure of *Recall*.

In the future, we plan to consider the social relationship of the user to further improve the performance. For applications, we can detect the hot and fine-grained locations and recommend them to users.

Acknowledgement. This work is supported by the Natural Science Foundation of China under Grant No. 61272240 and 61103151, the Doctoral Fund of Ministry of Education of China under Grant No. 20110131110028, the Natural Science Foundation of Shandong Province under Grant No. ZR2012FM037, the Excellent Middle-Aged and Youth Scientists of Shandong Province under Grant No. BS2012DX017 and the Fundamental Research Funds of Shandong University.

References

1. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: A content-based approach to geo-locating twitter users. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 759–768. ACM (2010)
2. Fink, C., Piatko, C., Mayfield, J., Finin, T., Martineau, J.: Geolocating blogs from their textual content. In: Working Notes of the AAAI Spring Symposium on Social Semantic Web: Where Web 2.0 Meets Web 3.0. AAAI Press (2009)
3. Amitay, E., Har’El, N., Sivan, R., Soffer, A.: Web-a-where: Geotagging web content. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 273–280. ACM (2004)

4. Backstrom, L., Kleinberg, J., Kumar, R., Novak, J.: Spatial variation in search engine queries. In: Proceeding of the 17th International Conference on World Wide Web, pp. 357–366. ACM (2008)
5. Serdyukov, P., Murdock, V., van Zwol, R.: Placing flickr photos on a map. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 484–491. ACM (2009)
6. Crandall, D.J., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world’s photos. In: Proceedings of the 18th International Conference on World Wide Web, pp. 761–770. ACM (2009)
7. Wing, B.P., Baldrige, J.: Simple supervised document geolocation with geodesic grids. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 955–964. Association for Computational Linguistics (2011)
8. Bigham, J.P., Sadilek, A., Kautz, H.: Finding your friends and following them to where you are. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, pp. 723–732. ACM (2012)
9. Sun, E., Backstrom, L., Marlow, C.: Find me if you can: Improving geographical prediction with social and spatial proximity. In: Proceedings of the 19th International Conference on World wide Web, pp. 61–70. ACM (2010)
10. Myers, S.A., Eunjoon, C., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090. ACM (2011)
11. Caverlee, J., Jeffrey, M., Cheng, Z.: Location prediction in social media based on tie strength. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 459–468. ACM (2013)
12. Dredze, M., Paul, M., Bergsma, S., Tran, H.: Carmen: A twitter geolocation system with applications to public health. In: AAAI Workshops (2013)
13. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, pp. 851–860. ACM (2010)

A Novel Calibrated Label Ranking Based Method for Multiple Emotions Detection in Chinese Microblogs

Mingqiang Wang¹, Mengting Liu¹, Shi Feng^{1,2}, Daling Wang^{1,2}, and Yifei Zhang^{1,2}

¹ School of Information Science and Engineering, Northeastern University

² Key Laboratory of Medical Image Computing (Northeastern University),

Ministry of Education, Shenyang 110819, P.R. China

{neuwangmingqiang, neuliumengting}@126.com,

{fengshi, wangdaling, zhangyifei}@ise.neu.edu.cn

Abstract. The microblogging services become increasingly popular for people to exchange their feelings and opinions. Extracting and analyzing the sentiments in microblogs have drawn extensive attentions from both academia researchers and commercial companies. The previous literature usually focused on classifying the microblogs into positive or negative categories. However, people's sentiments are much more complex, and multiple fine-grained emotions may coexist in just one short microblog text. In this paper, we regard the emotion analysis as a multi-label learning problem and propose a novel calibrated label ranking based framework for detecting the multiple fine-grained emotions in the Chinese microblogs. We combine the learning-based method and lexicon-based method to build unified emotion classifiers, which alleviate the sparsity of the training microblog dataset. Experiment results using NLPCC 2014 evaluation dataset show that our proposed algorithm has achieved the best performance and significantly outperforms other participants' methods.

Keywords: Microblog, Sentiment Analysis, Calibrated Label Ranking.

1 Introduction

With the advent of Web 2.0 technology, the mushrooming social media with user-generated-content have drawn extensive attentions from all over the world. Nowadays, people are willing to express their feelings and emotions via the microblogging services, such as Twitter and Weibo, because of easy accessibility and convenience. Therefore, the microblog has aggregated huge amount of tweets that contain people's rich sentiments. Extracting and analyzing the sentiments in microblogs has become a hot research topic for both academic communities and commercial companies.

A lot of papers have been published for analyzing the sentiments in blogs, reviews and news articles. However, there are several critical new challenges for detecting the emotions in microblogs.

(1) **Short text.** The microblog usually has a length limitation of 140 characters, which leads to extremely sparse vectors for the learning algorithms. The free and informal writing styles of users also set obstacles for the emotion detection in microblogs.

(2) **Fine-grained emotions.** Different from the traditional binary classification problem, namely classifying the text into *positive* or *negative* categories, the task of emotion detection needs to identify people’s fine-grained sentiments in microblogs, such as *happiness, sadness, like, anger, disgust, fear, and surprise*.

(3) **Mixed emotions.** Although the text is short, multiple emotions may be coexisting in just one microblog. Take the following tweet as an example, the twitterer expresses a *like* emotion as well as an *anger* emotion simultaneously in one tweet. Therefore, for the emotion detection task, each microblog can be associated with multiple emotion labels, which is rarely studied in the previous literature.

“@User: *What a fantastic movie! But the dinner sucks!!!!!!*”

To tackle these challenges, in this paper we regard the multiple emotions detection in microblog as a multi-label learning problem and propose a novel calibrated label ranking based method for classifying the fine-grained sentiments in the Chinese microblogs. Firstly, we transform the multi-label training dataset into single-label dataset. Then we propose a two-stage method that combining emotion lexicon and SVM to identify whether an instance (microblog or sentence) has emotion. Thirdly we construct Naïve Bayes (NB) and SVM multi-class classifiers to learn the confidence ranking value of the k emotion labels. Fourthly, we build k binary classifiers to update the ranking value and determine the threshold t . Finally we use the emotion lexicon to update the ranking value to get the final label order. For the ranking value of each label, if it is bigger than the threshold t , we put it into the relevant label set. Otherwise we put it into the irrelevant label set. To summarize, the main contributions of this paper are as follows.

(1) We propose a novel calibrated label ranking based framework for detecting the multiple fine-grained emotions in the Chinese microblogs.

(2) We combine the learning-based method and lexicon-based method to build unified emotion classifiers, which alleviate the sparsity of training microblog dataset and successfully improve the performance of the emotion classification results.

(3) Experiment results using NLPCC 2014 evaluation dataset show that our proposed algorithm significantly outperforms other team’s methods, and achieve the best performance in both the “Weibo Emotion Classification” and “Emotion Sentence Identification and Classification” subtasks.

The rest of the paper is organized as follows: Section 2 introduces the related work. In Section 3, we propose our calibrated label ranking based method for multiple emotions detection in Chinese microblogs. In Section 4, we introduce the experiment setup and results. Finally we conclude the paper and give the future work in Section 5.

2 Related Work

Our work is related to two directions: multi-label learning and sentiment analysis. Multi-label learning methods originally focused on the polysemy problem of text categorization [1-2]. After years of development, the multi-label learning algorithms have been widely used in the information retrieval [3-5], personalized recommendation

[6], Web mining [7], and etc. It is a hot research topic of the scholars. Zhang et al. gave a comprehensive review of the multi-label learning algorithms in [8]. Johannes et al. proposed a problem transformation algorithm “calibrated label ranking” that could be viewed as a combination of pairwise preference learning and the conventional relevance classification technique, where a separate classifier was trained to predict whether a label was relevant or not [9]. Another representative problem transformation algorithm “binary relevance” proposed by Matthew et al. transformed the multi-label classification problem to the binary classification problem [10]. Some other algorithm adaption methods were also widely used in the multi-label learning area. This kind of methods improved the traditional supervised machine learning algorithms to solve the multi-label problem. Zhang M-L et al. proposed the MLKNN improved by the lazy learning algorithm KNN to deal with the multi-label problem [11]. Elisseeff et al. improved the kernel learning algorithm SVM to fit the algorithm to the multi-label data [12].

The sentiment analysis researches can be traced back to early 2000’s. Pang and Lee [13] verified the effectiveness of applying machine learning techniques to sentiment classification, Costa [14] proposed a framework to integrate mining algorithms and software agent for building blog based sentiment applications. Zhang [15] provided a synthetic method to extract sentiment features from product reviews as product weakness finder. Huang [16] utilized the semantic information to construct a semantic sentiment space model to facilitate sentiment classification task.

Although a lot of papers have been published for multi-label learning and sentiment analysis, little work is done for detecting the fine-grained multi-label emotions in the microblog short text. In this paper, we propose a calibrated label ranking based method combined with emotion lexicon for the multiple emotions detection in Chinese Microblog.

3 Calibrated Label Ranking Based Emotion Classification

In the above section, we introduced the related work of multi-label learning and sentiment analysis. In this section, we propose a novel calibrated label ranking based method for sentence and microblog multiple emotions classification. Suppose $Y=\{y_1, y_2, \dots, y_k\}$ represents the emotion label set $\{happiness, sadness, like, anger, disgust, fear, surprise\}$ and $k=7$. Give the training dataset $D=\{(x_i, Y_i) | 1 \leq i \leq n\}$ where n is the number of instances in D and $Y_i \subset Y$, our task is to build a multi-label classifier that can predict the emotion label set $h(x) \subseteq Y$ of a new instance x . The overall framework of our proposed approach is shown in Figure 1.

In Figure 1, the proposed calibrated label ranking method has the following main steps. (1) We transform the multi-emotion label training microblog dataset into single-label dataset. (2) We build an emotional/non-emotional classifier to identify the emotional sentence in the testing dataset. (3) We propose a novel calibrated label ranking based method that integrating SVM and NB classifiers to get the ranking value of each emotion label of the sentence. (4) We construct k binary classifiers to update the ranking value and get the threshold. (5) We leverage the emotion lexicon to further update the ranking value and finally get the relevant label set by comparing the ranking value of each label with the threshold. In the following sections, we will discuss the steps of our proposed method in detail.

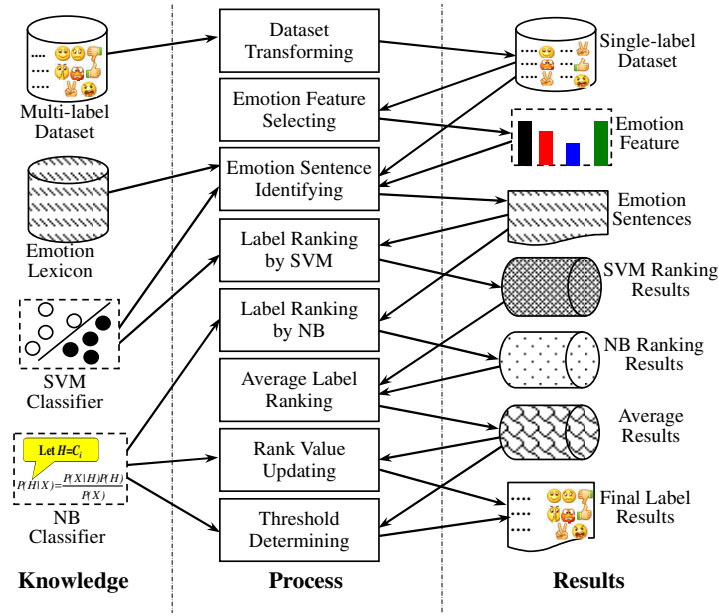


Fig. 1. The overall framework for calibrated label ranking based emotion classification

3.1 Training Dataset Transformation and Feature Selection

One popular strategy of solving the multi-label learning problem is to transform it into other well-established learning scenarios. For the given training dataset, each sentence in the microblog may contain one or two relevant emotion labels, therefore we need to transform the multi-label training dataset into the single-label train dataset so that we can use the traditional supervised machine learning algorithm to address the multi-label learning problem. The dataset transformation method is shown in Algorithm 1.

Algorithm 1. Training dataset transformation

Input: The multi-label training microblog dataset D ;

Output: The single-label training dataset D' after transformed;

Description:

1. FOR every microblog m in the multi-label training dataset
 2. FOR every sentence x_i in microblog m , where Y_i is the relevant emotion label set of the sentence x_i and $D = \{(x_i, Y_i) | 1 \leq i \leq n\}$, where n is the number of sentences in microblog m
 3. FOR every label y_j in the label set Y , where $Y = \{like, happiness, sadness, disgust, anger, fear, surprise\}$
 4. If $y_j \in Y_i$ put the training instance (x_i, y_j) into D'
-

In Algorithm 1, we associate each training item with only one emotion label, which transforms the multi-label dataset into a single-label dataset. Note that a sentence x in D may appear multiple times in the dataset D' .

Feature selection is the critical step for the text classification problem. In this paper, we select the words whose frequency is bigger than 2, which could alleviate the dimension disaster. That is to say, we ignore the words that appear less than twice in all the sentences. In addition, we select the emotion features listed in Table 1, which are potential good emotion indicators. Here we choose the punctuations, emoticons, emotion words extracted from the emotion ontology of Dalian University of Technology (DLUT) and the cyber words. Generally speaking, a sentence contains continuous punctuation always has an emotion. The emotion words and the emoticon can obviously imply that a sentence contains emotions. Some cyber words are also good symbols for people's emotions. We add these features into the user dictionary of Chinese segmentation tool, so that the tool can recognize these words and symbols.

Table 1. The Chinese Microblog Features

Feature Name	Feature Description and Examples
Punctuation	!, ?, ???, !!!, ?????!!!!
Emoticon	
Emotion word	Extracted from the DLUT emotion ontology
Cyber word	给力 (awesome), 稀饭 (like)

3.2 Multi-label Emotion Analysis of Weibo Sentence

To detect the multiple emotions in the microblog sentences, firstly we need to identify whether the candidate sentence is emotional or not. Then we need to detect the top 2 emotions of an emotional sentence. The target fine-grained emotion label set $Y=\{like, happiness, sadness, fear, surprise, disgust, anger\}$, and if an emotional sentence contains only one emotion, then the second emotion is set to be *none*.

3.2.1 A Two-Stage Method for Emotional Sentence Identification

In this section, we proposed a two-stage algorithm that combining the lexicon based and learning based methods for emotional sentence identification. The overall procedure of the proposed identification algorithm is shown in Figure 2.

We can see from Figure 2 that in the first stage, we utilize emotion lexicon to identify the emotional sentences in the testing dataset. If at least one word of the sentence is matched in the lexicon, we regard the sentence as emotional. In the second stage, we construct a SVM based classifier to identify the emotional sentences in the resting dataset that has no obvious emotion words. The aim of this two-stage approach is that we improve the Recall of the identification as well as keeping the Precision. Note that in the close setting, we use the DLUT emotion lexicon and in the open setting, we integrate emoticons, cyber words and extracted emotion words with DLUT lexicon.

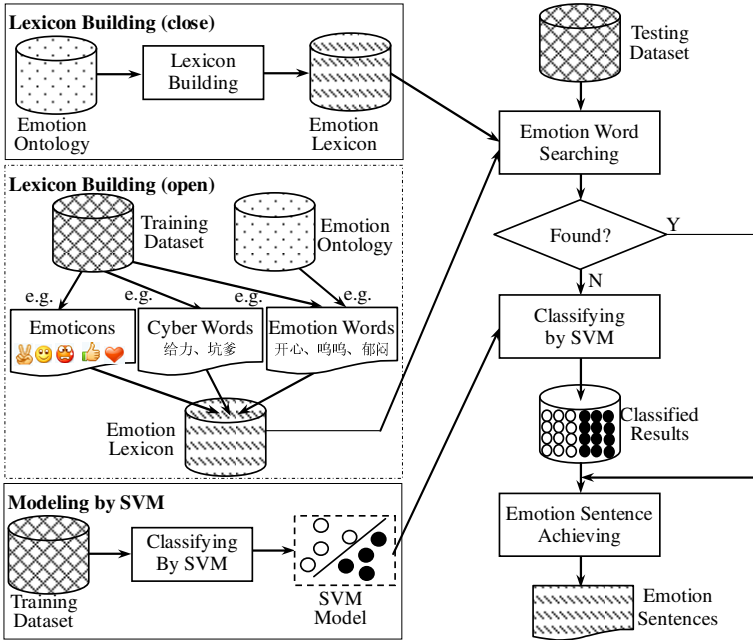


Fig. 2. The two-stage algorithm for emotional sentence identification

3.2.2 Calibrated Label Ranking

We leverage the transformed single-label training dataset to build a classifier $f(\cdot)$, by which we can get the confidence that an emotion is the proper label with a sentence. In this paper, $f(x_i, y_j)$ represents the confidence that sentence x_i has the emotion label y_j . Here $f(x_i, y_j)$ can be transformed to a rank function $rank_f(x_i, y_j)$. This rank function reflects all the real value output $f(x_i, y_j)$, where $y_j \subset Y$ and $1 \leq j \leq q$ ($q=7$). We use the Formula 1 to get the rank value.

$$rank_f(x_i, y_j) = \sum_{k=1, j \neq k}^7 \llbracket f(x_i, y_j) > f(x_i, y_k) \rrbracket \tag{1}$$

where $\llbracket \pi \rrbracket = 1$ if the predicate π holds, and 0 otherwise.

In this paper, we utilize the SVM and NB respectively to realize the label ranking. NB classifier can directly get the confidence of the k emotions. But SVM is a binary classifier, so here we use the 1-v-1 method to do the multi-class classification. In our experiment we adopt the average rank value $rank_f^{avg}(x_i, y_j)$ of NB and SVM.

3.2.3 Threshold Calibration

After we get the rank values of all the emotion labels, the key issue is to determine a threshold to separate the rank value list into relevant label set and irrelevant label set. In this paper we utilize the “binary relevance” algorithm to determine the threshold. The processing procedure is as follows:

- (1) Transform the multi-label training dataset into k ($=7$) separate datasets and each dataset is associated with a unique emotion label.
- (2) Construct k binary classifiers respectively corresponding with the k emotion labels using the training datasets.
- (3) Employ the k binary classifiers to update the threshold and the rank value for every unseen sentence.

Binary Classifier Construction. For every binary classifier $g_j(x_i)$, we firstly construct the train dataset D_j , $j \in \{1, 2, \dots, k\}$, where j denotes the emotion label and x_i represents a sentence. The original training dataset is denoted by $D = \{(x_i, Y_i) | 1 \leq i \leq m\}$, where m is the number of sentences in D and Y_i is the relevant label set of the sentence x_i . We use the Formula 2 to construct the training set of the i th classifier.

$$D_j = \{(x_i, \phi(Y_i, y_j)) | 1 \leq i \leq m\} \quad (2)$$

$$\phi(Y_i, y_j) = \begin{cases} +1, & y_j \in Y_i \\ -1, & \text{otherwise} \end{cases}$$

The separate training datasets D_j ($1 \leq j \leq k$) are utilized to train k binary classifiers $g_j(x_i)$ ($1 \leq j \leq k$) to classify every sentence in the testing dataset. For $g_j(x_i)$, if x_i is corresponding with y_j , then $g_j(x_i) = +1$, otherwise $g_j(x_i) = -1$. This is obviously a binary classification problem.

Rank Value Updating and Threshold Calibration. We use the following rule to update the rank value of label y_j $rank_f(x_i, y_j)$ and the threshold $rank_f^*(x_i, y_v)$ where y_v is a virtual label to separate the rank list.

$$rank_f^*(x_i, y_j) = rank_f^{avg}(x_i, y_j) + \llbracket g_j(x_i) > 0 \rrbracket \quad (3)$$

$$rank_f^*(x_i, y_v) = \sum_{j=1}^7 \llbracket g_j(x_i) < 0 \rrbracket \quad (4)$$

In the following section, we will further update the rank value using the emotion lexicon, and build the final multi-label classifier $h(x)$.

3.2.4 Updating Ranking Value by Emotion Lexicon

The emotion words are good indicators for emotion classification. If a sentence contains an emotion word of the label y_j , we can infer that this sentence has high probability to be associated with label y_j . Suppose E represents an emotion lexicon that has k ($=7$) categories, and each of the categories is associated with a label in $\{happiness, sadness, like, anger, disgust, fear, surprise\}$. For sentence x_i , if a word of x_i is matched in the lexicon E with label y_j , we utilize the Formula 5 to update the ranking value of y_j for x_i .

$$rank_f^*(x_i, y_j) = rank_f^*(x_i, y_j) + 1 \quad (5)$$

Finally based on the above formulas, we get the multi-label emotion classifier $h(x_i)$:

$$h(x_i) = \{y_j \mid rank_f^*(x_i, y_j) > rank_f^*(x_i, y_v), y_j \in Y\} \tag{6}$$

In Formula 6, the size of the relevant label set $h(x_i)$ may be bigger than two. In this situation we just select the top two emotions to fit with the evaluation task of the testing dataset. And another exception is that the relevant label set $h(x_i)$ may be empty, in this case we just select the top one emotion in the ranking list.

3.3 Multi-label Emotion Analysis of Weibo Text

The Weibo text usually consists of a group of sentences. The sentence level multi-label emotion analysis method can be seen as the basis of Weibo text level multiple emotion detection. So we can use the sentence level multi-label classification result in Section 3.2 to achieve the emotion labels at Weibo text level. Let x_i represent the sentence in Weibo m , where $1 \leq i \leq n$ and n is the number of sentences in m . We set the dominating emotion of the sentence the rank value 2, and 1 for the secondary emotion. Calculate the rank value $rank_m(m, y_j)$ of all the labels in Weibo m by the relevant label set of all the sentences in the Weibo.

$$rank_m(m, y_j) = \sum_{i=1}^n rank_m^{emo}(x_i, y_j) \tag{7}$$

where $rank_m^{emo}(x_i, y_j)$ is defined as follows:

$$rank_m^{emo}(x_i, y_j) = \begin{cases} 0, & y_i \text{ is the irrelevant label} \\ 1, & y_i \text{ is the secondary label} \\ 2, & y_i \text{ is the dominant label} \end{cases} \tag{8}$$

Similar to the sentence level multi-label emotion analysis, we employ the emotion lexicon to update the rank value of each label to get the final rank. After that we need to determine a threshold to separate the rank value list into relevant label set and the irrelevant label set. Here we defined $rank_m(m, y_v)$ as the threshold, where y_v is a virtual label. The threshold is calculated by counting the “none” labels of each sentence as follows.

$$rank_m(m, y_v) = \sum_{i=1}^n rank_m^{none}(x_i, y_v) \tag{9}$$

where y_v means the “none” label, and

$$rank_m^{none}(x_i, y_v) = \begin{cases} 0, & x_i \text{ has two emotions} \\ 1, & x_i \text{ has only one emotions} \\ 3, & x_i \text{ is not an emotional sentence} \end{cases} \tag{10}$$

Most sentences only contain one dominant emotion, which can easily cause the threshold $rank_m(m, y_V)$ bigger than the rank value $rank_m(m, y_j)$. We set a parameter α to balance the $rank_m(m, y_V)$. So the relevant label set $h(m)$ of Weibo m is calculated as:

$$h(m) = \{y_j \mid rank_m(m, y_j) \geq rank_m(m, y_V) - \alpha, y_j \in Y\} \quad (11)$$

If the rank value of a label is bigger than the threshold, we put the label into the relevant label set. Otherwise we put it into the irrelevant label set. As the same as the sentence level emotion analysis, if there are more than two emotion labels in the relevant label set, we just select the top two emotions. And if the relevant label set is empty, we regard the Weibo text as a non-emotional. In this paper, we empirically set $\alpha=2$ in the following experiment section. The tuning of parameter α is omitted due to length limitation.

Besides using the Formula 11, we can also consider the whole Weibo text as a long sentence and utilize the classification method proposed in Section 3.2 to detect the multiple emotions in the microblogs.

4 Experiment

4.1 Experiment Setup

We conduct our experiment on the real-world dataset that provided by NLPCC 2014 Emotion Analysis in Chinese Weibo Texts (EACWT) task¹. The EACWT training dataset contains 14,000 Weibo and 45,421 sentences. There are 6,000 Weibo and 15,693 sentences in the testing dataset of EACWT. We conduct experiments using a PC with Inter Core i7, 8 GB memory and Windows 7 as the operating system.

For the sentence level emotion analysis, we firstly transform the multi-label training dataset into single-label dataset using Algorithm 1. And then we identify whether a sentence is emotional or not using the two-stage method. We employ the transformed training dataset to construct the 7-classes classifier separately by NB and SVM to get the average ranking value of each label of an emotional sentence. Next we construct 7 binary NB classifiers to determine the threshold and update the rank value. At last we use the emotion lexicon to update the rank value of each label. For Weibo text level emotion analysis, we learn the Formula 11 to detect the multiple emotions. Another alternative strategy is that we can regard the whole Weibo text as a long sentence and employ the algorithm in Section 3.2 to classify the multiple emotions in the Weibo text.

4.2 Experiment Results

There were 7 teams participated the EACWT task of NLPCC 2014. We compare our results with other participators using the Average Precision (AP).

¹ http://tcci.ccf.org.cn/conference/2014/pages/page04_eva.html

4.2.1 Emotion Sentence Identification and Classification

In the close evaluation of the *Emotion Sentence Identification and Classification*, we strictly use the given training dataset and construct the emotion lexicon by the DLUT emotion ontology. In the open evaluation, we combine the dataset of NLPCC 2013, the sample and training dataset of NLPCC 2014 to form the new training dataset. We also manually add more emotion words into the emotion lexicon.

Emotion Sentence Identification. Firstly, we evaluate our proposed two-stage method for emotion sentence identification, i.e. classifying the sentences into emotional and non-emotional categories. The results are shown in Table 2.

In Table 2, all the three methods have similar F-Measures. However, the two-stage method (Lexicon+SVM) can achieve an extreme high Recall value. That is to say, most of the emotional sentences are included in the identification results, which pave the way for the further multi-label emotion classification steps.

Table 2. The comparisons for emotion sentence identification (open)

	Precision	Recall	F-Measure
SVM	49.24	57.73	53.15
Lexicon	40.49	77.14	53.11
Lexicon+SVM	38.41	87.21	53.40

Emotion Sentence Classification. Our approach combines the learning based method NB and SVM with emotion lexicon to get the ranking value of each label. To evaluate the effectiveness of our proposed method, we apply NB, SVM, NB+SVM, NB+Lexicon, SVM+Lexicon respectively to get the rank value of each label. In Table 3, CLR represents the Calibrated Label Ranking based method proposed in this paper. NEUDM-1 and NEUDM-2 denote the submit versions for the evaluation that are slightly different in feature selection methods. We can see from Table 3 that the combined approach CLR achieves the best performance compared with other methods. This validate that the proposed ensemble algorithm that combining NB, SVM and emotion lexicon is effective for sentence level multiple emotion detection.

Table 3. The comparisons for sentence level emotion classification (Average Precision, Strict)

	Result ID	NB	SVM	NB+ SVM	NB+ Lexicon	SVM+ Lexicon	CLR
Close	NEUDM-1	0.4479	0.4660	0.4778	0.4704	0.4883	0.5042
	NEUDM-2	0.4421	0.4580	0.4700	0.4578	0.4786	0.4911
Open	NEUDM-1	0.4772	0.4850	0.5044	0.4963	0.5128	0.5330
	NEUDM-2	0.4773	0.4850	0.5044	0.4953	0.5133	0.5317

In Table 4, we compare the sentence level emotion classification results with other participators of NLPCC 2014 EACWT task. The AVG and MAX represent the average and max value of all the participators. The Sec-Best denotes the second best value of all the participators. We can see that our proposed method significantly outperforms other participators' methods in all the evaluation settings.

Table 4. The comparison with other participators for sentence level classification

Result ID	Close		Open	
	AP(loose)	AP(strict)	AP(loose)	AP(strict)
NEUDM-1	0.5502	0.5042	0.5799	0.5330
NEUDM-2	0.5381	0.4911	0.5785	0.5317
AVG	0.3981	0.3666	0.5158	0.4791
Sec-Best	0.3032	0.2831	0.5489	0.5175
MAX	0.5502	0.5042	0.5799	0.5330

4.2.2 Weibo Emotion Classification

In the close evaluation of Weibo emotion classification, we strictly use the given training dataset and construct the emotion lexicon by the DLUT emotion ontology. We regard the whole Weibo text as a long sentence and adopt the calibrate label ranking algorithm proposed in Section 3.2 to detect the multiple emotions in Weibo text. In the open evaluation, we make use of the open classification result of the emotion sentence classification and employ the Formula 11 to realize the Weibo text level emotion classification. In Table 5, we compare our results with the valid results of other participators of NLPCC2014 EACWT task.

Table 5. The comparison with other participators for Weibo level classification

Result ID	Close		Open	
	AP(loose)	AP(strict)	AP(loose)	AP(strict)
NEUDM-1	0.6033	0.5115	0.5851	0.4960
AVG	0.4306	0.3678	0.5145	0.4406
Sec-Best	0.5736	0.4756	0.5309	0.4668
MAX	0.6033	0.5115	0.5851	0.4960

In Table 5, our proposed method has achieved the best performance in all the participators. Note that in NEUDM-1, the AP values of the close setting are better than the AP values of the open setting, namely regarding the Weibo text as separate sentences does not lead to a better performance than regarding the whole Weibo text as a long sentence. This may be because of the accumulation of errors at the sentence level emotion classification. The Formula 11 is counting on the classification result of each sentence in the Weibo text. Therefore, combining each sentence' top ranked two labels may accumulate more errors. We can tackle this problem by regarding the whole Weibo text as a long sentence for emotion analysis.

5 Conclusions and Future Work

Recently, emotion analysis in microblogs has become a hot research topic, which aims to classify the fine-grained sentiments in the short text. Different from the traditional sentiment analysis problem, the fine-grained sentiments co-exist with each other in the

short text and the emotion detection in microblog can be regarded as a typical multi-label learning problem. In this paper, we proposed a novel calibrated label ranking based multi-label emotion analysis approach for Chinese microblogs. Our proposed model combined multi-label learning algorithm and emotion lexicon together, which provided a comprehensive understanding of the embedded emotions in the short text. Experiment results based on NLPCC 2014 EACWT evaluation task showed that our proposed method significantly outperformed the methods of other participants and our proposed ensemble learning framework achieve better performance than other single classifiers or other combined approaches.

Our future work includes further improving the performance the emotional sentence identification task, which is also a critical step for the evaluation task. We intend to automatically find more effective emotion features and integrate more classifiers. We also want to propose new multi-label learning methods for the multiple emotions detection task. We will study on measuring the fine-grained emotion similarity between the Chinese microblogs and leverage the MLKNN based lazy learning method for multiple emotions detection.

Acknowledgements. This work is supported by the State Key Development Program for Basic Research of China (Grant No. 2011CB302200-G), State Key Program of National Natural Science of China (Grant No. 61033007), National Natural Science Foundation of China (Grant No. 61100026, 61370074, 61402091), and Fundamental Research Funds for the Central Universities (N120404007).

References

1. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 1–13 (2007)
2. Tsoumakas, G., Zhang, M., Zhou, Z.: Learning from multi-label data. Tutorial at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2009), Bled, Slovenia (2009)
3. Schapire, R., Singer, Y.: BoosTexter: A boosting-based system for text categorization. *Machine Learning* 39(2-3), 135–168 (2000)
4. McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: Working Notes of the AAAI 1999 Workshop on Text Learning, Orlando, FL (1999)
5. Ueda, N., Saito, K.: Parametric mixture models for multi-label text. In: Becker, S., Thrun, S., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pp. 721–728. MIT Press, Cambridge (2003)
6. Song, Y., Zhang, L., Giles, L.: A sparse Gaussian processes classification framework for fast tag suggestions. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pp. 293–302. Napa Valley, CA (2008)
7. Tang, L., Rajan, S., Narayanan, V.: Large scale multi-label classification via metalabeler. In: *Proceedings of the 19th International Conference on World Wide Web (WWW 2009)*, Madrid, Spain, pp. 211–220 (2009)
8. Zhang, M., Zhou, Z.: A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8), 1819–1837 (2014)

9. Johannes, F.: Multi-label classification via calibrated label ranking. *Machine Learning* 73(2), 133–153 (2008)
10. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recognition* 37(9), 1757–1771 (2004)
11. Zhang, M., Zhou, Z.: ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7), 2038–2048 (2007)
12. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pp. 681–687. MIT Press, Cambridge (2002)
13. Pang, B., Lee, L.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: *Proc. 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86 (2002)
14. Costa, E., Ferreira, F., Brito, P., et al.: A framework for building web mining applications in the world of blogs: A case study in product sentiment analysis. *Expert Systems with Applications* 39(4), 4813–4834 (2012)
15. Zhang, W., Xu, H., Wan, W.: Weakness Finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis. *Expert Systems with Applications* 39(9), 10283–10291 (2012)
16. Sui, H., You, J., Zhang, J., Zhang, H., Wei, Z.: Sentiment Analysis of Chinese Micro-blog Using Semantic Sentiment Space Model. In: *Proceedings of 2nd International Conference on Computer Science and Network Technology, ICCSNT 2012*, pp. 1443–1447 (2012)

Enhance Social Context Understanding with Semantic Chunks

Siqiang Wen, Zhixing Li, and Juanzi Li

Dept. of Computer Sci. and Tech., Tsinghua University, China
{wensq2329,adam0730,lijuanzi2008}@gmail.com

Abstract. Social context understanding is a fundamental problem on social analysis. Social contexts are usually short, informal and incomplete and these characteristics make methods for formal texts give poor performance on social contexts. However, we discover part of relations between importance words in formal texts are helpful to understand social contexts. We propose a method that extracts semantic chunks using these relations to express social contexts. A semantic chunk is a phrase which is meaningful and significant expression describing the fist of given texts. We exploit semantic chunks by utilizing knowledge learned from semantically parsed corpora and knowledge base. Experimental results on Chinese and English data sets demonstrate that our approach improves the performance significantly.

1 Introduction

Social context understanding, whose aim is to get the main idea of the given social contents, has become more and more important and fundamental problem with the explosive growth of user generated content. However, social contexts are usually short, informal and incomplete, these characteristics make us difficult to extract phrases to express the given social contents. Thus, it is a challenging problem in social media processing.

Keyphrase extraction can be regarded as an aspect of text understanding. For formal texts, amount of effective approaches have been proposed and nice results have been achieved. Keyphrase extraction approaches can be roughly categorized into supervised [3,17] and unsupervised [15,11,7]. Supervised methods use various kinds of features to build a classifier. There are litter works using supervised methods for keyphrase extraction for social contexts, as lack of manually annotated training data. Unsupervised algorithms, regarding keyphrase extraction as a ranking task, utilize TF-IDF [15], co-occurrence[11], topic[7] and so on to rank candidates. Based on unsupervised approaches, a number of researchers have used them on twitter's texts. [18,19,1] have used and expanded TF-IDF and TextRank for keyphrase extraction. However, performance of these methods on social contexts is not as good as that on formal texts. Social contexts are usually short to record some trivia and their structure and grammar are usually incomplete. In addition, social contexts contain lots of impurities

like abbreviations, misspelled words, slang words, and emoticons. These characteristics of social contexts make keyphrase extraction in difficulty to deal with social contexts. These methods can only extract words and simple phrases and the precision is very low. For example, the average number of Chinese character of Sina keywords [6,16] is about 2.08. Many meaningful and semantic phrases can't be extracted with these methods.

Social messages are casual logs of users' everyday life, which often lack of structure compared to formal text such as book and news reports; nevertheless there are some complete words and phrases in social contexts and most of these words and phrases are related to main idea of social contexts. Machine can't understand such texts, but human can capture the gist of social contexts by using part of complete words and phrases. Our method is just based on the idea that we usually do not need all relations between words like parser and only part of relations between importance words are enough to capture the main idea of the given sentence. Therefore, we can use such relations to extract semantic chunks to help to understand social contexts.

In this paper, we denote a semantic chunk as a phrase which is meaningful and significant expression describing main idea of given texts. Semantic chunk consists of semantic dependency words, which may be not consecutive words. For example, given texts *most of the passengers on board survived*, we will get semantic chunk such as "*passengers survived*", in which "*passengers*" and "*survived*" are not consecutive words, in other word there are several words between "*passengers*" and "*survived*".

To acquire semantic chunks, there are several problems to solve, such as how to find important relations, how to use these relations to form readable semantic dependency phrase and how to solve the limit of labeled corpora. To solve these problems, we use an corpus with annotated semantic dependency relationships for formal language to obtain dependent knowledge between words of nouns, verbs and adjectives. Then, we extract semantic chunks from social content by incorporating these dependency relationships with a knowledge base, WordNet for English and Tongyici Cilin for Chinese. Specifically, we propose a model which identifies the important words that evoke a semantic phrase in a given sentence and the dependent words which are dependent on the target words. The model is trained on semantic dependency corpora and extended by a external knowledge base since semantic dependency corpora is limited for social content. We acquire some phrases as candidates at first and we use chunk knowledge learnt by section 4.1 and some rules to expand candidates to form semantic chunks.

The main contributions of our paper are summarized as follows.

- We propose the method to use semantic chunks to solve the problem of social content understanding. We utilize part of semantic dependency relations between importance words learned from semantic dependency corpora and knowledge base to extract semantic chunks to capture the gist of the given document. The method does not need all relations between words like parser.

- We learn word knowledge, relation knowledge and distance knowledge from semantic dependency corpora. With these knowledge, we can extract long distance dependency phrases to form semantic chunks.
- To verify the effectiveness and efficiency of our method, we evaluated our method over Chinese and English social contexts, and the experimental results show that our method significantly outperforms the baseline methods.

2 Related Work

As this paper mainly studies social context understanding by extraction semantic chunks from them, we focus our literature review for approaches about keyphrase extraction and dependency parsing.

Keyphrase extraction is the process that identify a few meaningful and significant terms that can best express contexts. Generally speaking, keyphrase extraction approaches can be roughly categorized into two principled approaches: supervised and unsupervised. Supervised algorithms consider keyphrase extraction as a classification problem to classify a candidate phrase into either keyphrase or not. [3,17] have used some features, such as frequency, location, statistical association and other linguistic knowledge, to classify a candidate phrase. Due to lack of manually annotated training data, researchers hardly use supervised methods for keyphrase extraction for social contexts.

Unsupervised algorithms usually regard keyphrase extraction as a ranking task, which assigns a score to each candidate phrases by various methods then picks out the top-ranked terms as keyphrases. [15] has proposed a simple unsupervised algorithms using TF-IDF to extract keyphrase. Graph-based ranking methods become popular after TextRank model [11] proposed by Mihalcea and Tarau. Some approaches have been proposed to improve TextRank. Liu proposed other unsupervised algorithms, including clustering-based [8] and topic-based [7] methods.

While existing research mainly focuses on formal articles, the rapid growth of social network raises the needs of research on informal language. Informal language contexts are much shorter than formal articles. It is more difficult to extract keyphrase from informal contexts than from traditional articles. [18] uses TFIDF and TextRank, two standard keyword ranking techniques, to extract keyword. NE-Rank [1] proposes an enhanced PageRank to extract keyphrase for Twitter. [19] modifies Topical PageRank [7] to find topic keyphrase. PolyU [14] extracts core words and expands the identified core words to the target keyphrases by a word expansion approach. These unsupervised methods can extract words and simple phrases, but can not identify more meaningful and semantic phrases. Words and simple phrases can't accurately cover the mean of texts.

Dependency parsing can provide a representation of lexical or semantic relations between words in a sentence and have been designed to be easily extract textual relations. Stanford Dependencies [9] provides English and Chinese dependent relations between words. But these parser can't perform nice on social

contexts. Factually, we usually do not need all relations between words in a given sentence, and we only want the relations between importance words. In this paper, we use dependent knowledge to get the relations between importance words, such nouns, verbs and adjectives. Parser need full structure information of sentence, but we need part of structure information.

3 Framework

The method in this paper is mainly inspired by the idea that semantic chunks in social contexts are helpful to understand social texts. We find candidate phrases and then rank them to select semantic chunks. Not all words in a document are fit to be selected as candidate phrases. In [17], candidate phrases were found using n-gram. Liu [8] used exemplars to extract multi-word candidate phrases. Mihalcea and Tarau in [11] used n-grams as a post-processing step to form phrases. However, in this paper, we use chunk knowledge to extract candidate phrases rather than words and then select semantic chunks. Our framework consists of two processes, including

- **Dependency Knowledge Learning** We learn semantic dependency knowledge from annotated semantic dependency corpus for formal texts. The knowledge contains word dependency knowledge, relation dependency knowledge and distance knowledge.
- **Semantic Chunking** Given a sentence, a set of semantic chunk is generated using learned knowledge and external knowledge bases. The step contains target words identification, semantic pair discovery and semantic chunk generation.

We now introduce semantic dependency corpora which are annotated with dependency grammar and some related denotations for they frequently are used below. We use Chinese and English semantic dependency corpora(SND [13] and TreeBank) to learn chunk knowledge. SDN is Chinese semantic dependency corpus built by Tsinghua University. We gain English semantic dependency corpus by Penn TreeBank with Stanford Dependencies. Table 1 shows some import figures about two semantic dependency sets we use in this paper. Pair knowledge is the number of word pairs whose two word have semantic dependency relation each other. Relation knowledge is the number of pairs (r_a and r_b).

Table 1. Statistics on Semantic Dependency corpora

<i>Data Set</i>	<i>Sentences</i>	<i>Words</i>	<i>Vocabulary</i>	<i>Pair knowledge</i>	<i>Relation knowledge</i>
SDN	132396	908048	34539	550536	619084
TreeBank	240873	2514549	62632	1316311	1846158

In semantic dependency datasets, each sentence is represented by a dependency tree. For example, Figure 1 shows the dependency tree of the sentence

The sales of newly launched iPhone_5s disappoint investor. The root is "disappoint". Two words have semantic dependency relation if there is a directed edge between the two words. For example, "investor" is dependent on "disappoint" and the two words have relation of "dobj" for an edge links the two words.

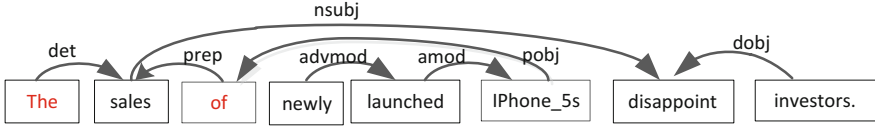


Fig. 1. A semantic dependency tree

We now give some related denotations used in this paper. Table 2 defines some variables used in our approach.

Table 2. Notation of some frequently occurring variables

Symbol	Description
d :	a document of social contexts, a sentence or several sentences, which can be represented as a sequence of words $(w_1, w_2, \dots, w_{n_d})$, where n_d means the number of words in the document d
W_K :	word vocabulary of semantic dependency corpora
Kno_w :	Word knowledge of semantic dependency corpora, which is represented as the set of $\{(w_i, w_j, r_a)\}$, where w_i and w_j has relationship of r_a
Kno_r :	Relation knowledge of semantic dependency corpora, which is represented as the set of relation sequential knowledge $(\{(r_a, r_b)\})$, which means r_a and r_b are two sequential co-occurring relationships.

4 Method

4.1 Dependency Knowledge Learning

In this part, we learn semantic dependency knowledge from semantic dependency datasets. The knowledge contains lexical collocations, and distance value between two dependent words and three words’s semantic relations.

For each sentence, if two words (w_i, w_j) have dependent relation (r_a) , we put (w_i, w_j, r_a) into Kno_w . There is a special case, for example, in Figure 1, "sales" and "iPhone_5s" have semantic dependency relation. "of" is in the dependent path ("iPhone_5s" \rightarrow "of" \rightarrow "sales"), but it is a red word. The red words are functional words and we remove the pair relations about functional words, because they are useless to capture meaning of content. So "sales" and "iPhone_5s" are dependent words due to transitivity. We use transitivity to remove form words and get expanded dependent words.

We extract words dependency knowledge($C_w(w_i, w_j)$), and distance knowledge($C_d(w_i, w_j)$). $C_w(w_i, w_j)$ is defined as follows:

$$\begin{aligned}
 C_w(w_i, w_j) &= p(w_i w_j) \lg \frac{p(w_i w_j)}{p(w_i)p(w_j)} \\
 &= \frac{\#N(w_i, w_j)}{\#Tp} \lg \frac{\#N(w_i, w_j)(\#Tw)^2}{\#N(w_i)\#N(w_j)\#Tp}
 \end{aligned} \tag{1}$$

where $\#N(w_i, w_j)$ is the number of pairs(w_i, w_j) occurred in the corpus, $\#N(w_i)$ is the number of w_i , $\#Tw$ is the number of words and $\#Tp$ is the number of pairs in the annotated corpus. w_i is word’s prototype and part-of-speech. $C_d(w_i, w_j)$ is the averaged distance between two dependent words in the corpus.

We expect that semantic chunk contains not only two words but also three or more words. If r_a and r_b have the same word in the dependency tree, we put (r_a, r_b) into $Knor$. For example, in Figure 1, *nsubj* and *dobj* both have word "disappoint", then we put $(nsubj, dobj)$ into $Knor$. In addition, we learn $(h(r_a, r_b))$ and other probability ($p(r|(w_i, w_j))$) for forming semantic chunk whose number of word is more than two. $p(r|(w_i, w_j))$ is the probability of relation r when given words w_i and w_j . $h(r_a, r_b)$ is mutual information co-occurrence of two relations (r_a, r_b) . In section 4.2, we use this knowledge and rule to get semantic chunks which are more than two words.

4.2 Semantic Chunking

We denote S_d as semantic chunks, which can be denoted as $(s_{d_1}, s_{d_2}, \dots, s_{d_{n_s}})$, and denote s_{d_i} as a word or phrase, which can be represented as a sequence of words $(w_{d_{i_1}}, w_{d_{i_2}}, \dots, w_{d_{i_n}} | d_{i_1} < d_{i_2} < \dots < d_{i_n})$ where n is among 1 and 3. A semantic chunk consists of target words and their relate words. Target word which can evoke a semantic phrase in a given sentence represents the dominant concepts in the social content. We denote T_d as target word set of d . s_{d_i} can be represented as follow.

$$s_{d_i} = \begin{cases} (w_i), & w_i \in T_d \\ (w_i, w_j), \exists r_a(w_i, w_j, r_a) \in Kno_w \\ (w_i, w_j, w_k), \exists r_a(w_i, w_j, r_a) \in Kno_w, \\ & \exists r_b(w_j, w_k, r_b) \in Kno_w, \\ & (r_a, r_b) \in Knor \end{cases} \tag{2}$$

Words(w_i, w_j, w_k) can be expanded by knowledge base, then determine whether (w_i, w_j, r_a) and (w_j, w_k, r_b) are in Kno_w . For example, w_i and w_j are expanded to w_{ik} and w_{jl} (4.2), where w_{ik} and w_{jl} are in W_K , if there is r_a and (w_{ik}, w_{jl}, r_a) is in Kno_w , then (w_i, w_j, r_a) is in Kno_w .

The task of semantic chunk extraction is to find a set of semantic chunks S_d when given a item of social contexts d . In other word, we find semantic dependency phrases from lexical collocations of all words in d . We denote the

number of lexical collocations is C_{all} . Semantic chunk candidates can be obtained with three steps, including target words identification semantic chunk discovery and semantic chunk generation.

Target Words Identification. The size of C_{all} is 2^n , where n is the length(the number of words) of social text d . The size of C_{all} is too large. C_{all} contains all combinatorial words. We use target word to reduce candidates without affecting the result. What's more, we think important target words are usually in semantic chunks. Generally speaking, verbs, nouns, adjectives, and even prepositions can evoke phrases under certain conditions. In [2], target words are identified by rules followed Johansson and Nugues [4], they select verbs, nouns, adjectives, and even prepositions as target words. Given a text, verbs, adjectives, adverbs and prepositions usually depend on nouns or be depended by noun. In other word, we only select noun and extract their related dependent relations, and thus we can get relevant verbs, adjectives and so on. So we pick out nouns as potential target words. Especially, entity is more important to dominant concept than other words. Therefore, we prefer proper nouns as goal words and give three bonus to proper nouns. We also consider frequency and position as features. We denote $score(w_i)$ as the score of target value. If w_i is target word, $score(w_i)$ is calculated by above strategy. If not, $score(w_i)$ equals one.

Semantic Pair Discovery. Given a sentence, in order to get semantic phrases, we first get target words in 4.2, then use these words, knowledge base and chunk knowledge to get related words set of target words. The chunk knowledge learnt from dependency corpus is limited. If we only use some part-of-speech patterns to select relate words of word that is not in W_K , most of the results are not readable and meaningless. Therefore, we utilize knowledge base, lexical collocations, part-of-speech knowledge and distance knowledge to get semantic chunk. We use **Tongyici Cilin** (A Dictionary of Syn-onyms) as knowledge base for Chinese and **WordNet** [12] for English.

Formally, we define wd as a window's size whose furthest word is distance target words as wd , where wd can be set the length of sentence. We find accompaniment words of t_i in a certain range of window(wd). We denote $Expand(w_i) = \{w_{i1}, w_{i2}, \dots, w_{ie}\}$ as a set expanded w_i by knowledge base with semantic category. Any word in $Expand(w_i)$ is in W_K . Let Sim_{ij} as similarity of w_i and w_{ij} . Sim_{ij} is calculated through knowledge base.

We define $R_w(w_i, w_j)$ as the word value and $R_d(w_i, w_j)$ as the distance value of w_i and w_j . $R_w(w_i, w_j)$ will be calculated as follow:

$$R_w(w_i, w_j) = \begin{cases} C_w(w_i, w_j), & w_i, w_j \in W_K \\ \sum_{k=1}^e \sum_{l=1}^e p_{kl} S(w_{ik}, w_{jl}), & else \end{cases} \quad (3)$$

where w_i is a target word. If w_i and w_j both are not in W_K , we will expand w_i and w_j by knowledge base. e is amount of expanded words, and $S(w_{ik}, w_{jl})$ is,

$$S(w_{ik}, w_{jl}) = C(w_{ik}, w_{jk}) \times Sim_{ik} \times Sim_{jl} \quad (4)$$

p_{kl} is,

$$p_{kl} = \frac{\#N(w_{ik}, w_{jk})}{\sum_{k=1}^e \sum_{l=1}^e \#N(w_{ik}, w_{jk})} \quad (5)$$

Suppose the distance between w_i and w_j is d_{ij} . $R_d(w_i, w_j)$ is the average distance in the corpus. If w_i and w_j both are not in W_K , we will expand w_i and w_j by knowledge base and use method like $R_w(w_i, w_j)$ to calculate $R_d(w_i, w_j)$. In order to determine whether w_i and w_j can form semantic chunk candidate, we then model the probability of candidate d as a function incorporating words knowledge and distance knowledge.

$$R(w_i, w_j) = \alpha R_w(w_i, w_j) + (1 - \alpha)(|R_d(w_i, w_j) - d_{ij}|)^{-1} \quad (6)$$

Semantic Chunk Generation. For each target words, we select top- m related words to form candidate set(EW), which is a sub set of C_{all} . We remove many unimportant candidates from C_{all} through above two steps. We can get semantic chunk candidates whose size is two. In this part, we use chunk knowledge learnt by section 4.1 and some rules to expand candidates.

Given three words(w_i, w_j, w_k), w_i and w_j have relation r_m , w_j and w_k have relation r_n . Then

$$C(w_i, w_j, w_k) = p(r_m|(w_i, w_j))p(r_n|(w_i, w_j))h(r_m, r_n) \quad (7)$$

We select top- v of all $C(w_i, w_j, w_k)$. Then selected phrase (w_i, w_j, w_k) as new candidates replaces (w_i, w_j) and (w_j, w_k). $R(w_i, w_j, w_k)$ is gotten by

$$R(w_i, w_j, w_k) = (R(w_i, w_j) + R(w_j, w_k))/2 \quad (8)$$

Then we select top n_s as semantic chunks from all candidates by standard logistic regression model. We use two features. One feature is score of phrases($R(w_i, w_j, w_k)$ or $R(w_i, w_j)$). Another is target value of phrases(section 4.2).

Through above steps, We get some two and triple phrases. Then we add some expanded words by heuristic rules to form phrases. Take *I have a beautify house in the woods* for example, we can extract "house woods", but it's not nice phrase. We need to make some change. We add preposition(*in*) to phrase "house woods". "house in woods" is better than "house woods".

5 Experiments

5.1 Datasets

As far as we know there is no existing benchmark dataset and no gold standard answers for semantic chunks on social contexts. To evaluate the performance

of our method, we carry out our experiments on two real world datasets, microblogs(Chinese) crawled from Sina Weibo(China) and news comments(English) from Yahoo!. The blogs contain blog posts that cover a diverse range of subjects. The statistics of the datasets is shown in Table 3, where $|D|$, $|W|$, $|V|$, $|N_s|$, $|N_w|$ are the number of document, the number of words, the vocabulary of contexts, the average number of sentences in each document and the average number of words in each sentence, respectively.

Table 3. Statistics on DataSets

<i>Dataset</i>	$ D $	$ W $	$ V $	$ N_s $	$ N_w $
Sina(cn)	1000	129304	15318	7.06	18.32
Yahoo!(en)	1000	97392	10821	5.96	16.34

We use some heuristic rules to filter out the noisy words in advance. Firstly we remove emoticons and URL. Secondly, English documents are tokenized and tagged, while Chinese documents are segmented into words and then tagged. Finally, for both datasets, we identify stop words. We do not remove stop words in the original texts for stop words may be used as expanded words.

5.2 Experiments Setup

Evaluation Methods. To guarantee the low noise of the manual annotated data, we totally employ 1000 blogs posts(Chinese) and 1000 news comments (English), and for each document we ask at least 3 different annotators to rate the corresponding semantic chunks and keyphrases. Finally, each document is rated by averaging ratings from annotators. For each document, annotators were asked to rate the labels based on the following ordinal scale [5]:

- 3:** Very good phrase, completely capturing gist of the document
- 2:** Reasonable and readable phrase, but not completely capturing gist
- 1:** Phrase is related to the contexts, but not readable
- 0:** Phrase is completely inappropriate

Baseline Methods. We use **SmanC** to denote semantic chunk with knowledge base and **SmanC-KB** to denote semantic chunk without knowledge base. We select two major types of baseline methods for comparison: unsupervised keyphrase extraction methods and parsing which is the process of analysing contexts.

Unsupervised Keyphrase Extraction Method: Unsupervised keyphrase extraction methods assigns a score to each candidate phrases by various methods then picks out the top-ranked terms as result. There are some unsupervised keyphrase extraction methods proposed for social contexts. In this paper, we use **TextRank** [11] as baselines. TextRank build a word graph based on the

co-occurrence between words, then execute PageRank on the graph to give score for each keyphrase candidate. [18,1] have used TextRank for extraction with social contexts. To build the graph for TextRank, we will select noun phrases, verb phrases and adjectives as candidates. According to our experiments, TextRank’s best score is achieved when vertices’ co-occur within a window of two words for microblog documents and three words for news comments. Damping factor of 0.85, uniform prior of 1.0 and 50 iterations are set.

Parsing: There are many parser proposed. In this paper, we use **Stanford Dependencies** [9] for English documents and **MSTParser**(Minimum-Spanning Tree Parser) [10] for Chinese documents as baselines. MSTParser is trained by SDN, a labeled semantic dependency corpora using in section 4.1. In our experiment, we will extract phrases whose one word is noun or verb and other words semantically depend or be depended by the word. We will extract noun phrases and verb phrases. Then final result is selected by frequency and position information, such as head or tail of sentence.

5.3 Results and Analysis

Table 4 gives the performance results on two datasets, and the best performances in the comparisons are highlighted in bold. *m* and *v* both are set to 2 according to experiment performance. As the length of each document is short, we select top-3 as final phrases for each methods. Our proposed method **SmanC** outperforms the baseline methods TextRank and Parsing on both datasets. Our method utilizes semantic dependency knowledge, and don’t care the complete structure of sentence. We can see textRank’s performance is not good , because document is very short and the co-occurrence relation can not reflect the meaning of whole document. In addition, textRank can only extract some simple words and phrases. These reasons make textRank worse than other methods.

Table 4. Overall results of various methods for social contexts

	<i>SmanC</i>	<i>SmanC-KB</i>	<i>parser</i>	<i>textRank</i>
Sina(cn)	2.237	2.156	1.918	1.424
Yahoo!(en)	1.871	1.859	1.548	1.213

Results of SmanC-KB is higher than that of parser for microblogs posts and news comments. The performance of SmanC is higher than SmanC-KB for two datasets. The reason is that SmanC utilizes knowledge base to expand candidate set and can extract more phrases. But we can find some expanded phrases not readable. SmanC can extract longer phrases than the other methods. The average length of phrases of SmanC is 4.46 for Chinese documents and 2.53 for English document. However, a small part of long phrases is not reasonable. The performance of Chinese documents is better than that of English documents, because we deal with words in English documents while through Chinese word

segmentation Chinese word are not only a word but a simple phrase. For example, the phrase "New York" has two words in English, but a word after segmentation in Chinese. Although we make use of simple rules to combine some nouns and verbs to form nice phrases in order to get better performance, the results of news comments are not as good as microblogs'.

Furthermore, we investigate the results and discover that our method failed to find and appropriate phrases when encountered wrong tag of words. For example, give a sentence *The/DT injured/JJ included/VBD two/CD FBI/NP agents/NNS* , and we only get *FBI agents*. In fact, the tag of word *injured* is NN. With the right tags, we will get better phrase *injured included FBI agents*. Sometimes, we will not get reasonable and readable phrase without proper tags.

6 Conclusion and Future Work

We extract semantic chunks from social contexts to solve the problem of social context understanding. There're many future directions of this work such as automatic labelling. We will explore our method on other languages and on other test data to investigate and validate the robustness of our approach.

Acknowledgment. The work is supported by 973 Program (No. 2014CB340504), NSFC (No. 61035004), NSFC-ANR (No. 61261130588), European Union 7th Framework Project FP7-288342 and THU-NUS NExT Co-Lab.

References

1. Bellaachia, A., Al-Dhelaan, M.: Ne-rank: A novel graph-based keyphrase extraction in twitter. In: Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, WI-IAT 2012, vol. 01, pp. 372–379. IEEE Computer Society, Washington, DC (2012)
2. Das, D., Chen, D., Martins, A.F., Schneider, N., Smith, N.A.: Frame-semantic parsing (2013)
3. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, pp. 216–223. Association for Computational Linguistics, Stroudsburg (2003)
4. Johansson, R., Nugues, P.: Lth: Semantic structure extraction using nonprojective dependency trees. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval 2007, pp. 227–230. Association for Computational Linguistics, Stroudsburg (2007)
5. Lau, J.H., Grieser, K., Newman, D., Baldwin, T.: Automatic labelling of topic models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 1536–1545. Association for Computational Linguistics, Stroudsburg (2011)
6. Liu, Z., Chen, X., Sun, M.: Mining the interests of chinese microbloggers via keyword extraction. *Frontiers of Computer Science* 6(1), 76–87 (2012)

7. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 366–376. Association for Computational Linguistics, Stroudsburg (2010)
8. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 1, pp. 257–266. Association for Computational Linguistics, Stroudsburg (2009)
9. de Marneffe, M.C., Manning, C.D.: The stanford typed dependencies representation. In: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, CrossParser 2008, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2008)
10. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 523–530. Association for Computational Linguistics, Stroudsburg (2005)
11. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP 2004, pp. 404–411 (2004)
12. Miller, G.A.: Wordnet: A lexical database for english. *Commun. ACM* 38(11), 39–41 (1995)
13. Mingqin, L., Juanzi, L., Zhendong, D., Zuoying, W., Dajin, L.: Building a large chinese corpus annotated with semantic dependency. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, SIGHAN 2003, vol. 17, pp. 84–91. Association for Computational Linguistics, Stroudsburg (2003)
14. Ouyang, Y., Li, W., Zhang, R.: 273. task 5. keyphrase extraction based on core word identification and word expansion. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval 2010, pp. 142–145. Association for Computational Linguistics, Stroudsburg (2010)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 513–523 (1988)
16. Liu, Z., Tu, C., Sun, M.: Tag dispatch model with social network regularization for microblog user tag suggestion (2012)
17. Turney, P.D.: Learning algorithms for keyphrase extraction. *Inf. Retr.*, 303–336 (2000)
18. Vu, T., Perez, V.: Interest mining from user tweets. In: Proceedings of the 22nd ACM International Conference on Conference on Information Knowledge Management, CIKM 2013, pp. 1869–1872. ACM, New York (2013)
19. Zhao, W.X., Jiang, J., He, J., Song, Y., Achananuparp, P., Lim, E.P., Li, X.: Topical keyphrase extraction from twitter. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 379–388. Association for Computational Linguistics, Stroudsburg (2011)

Estimating Credibility of User Clicks with Mouse Movement and Eye-Tracking Information*

Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma

¹ State Key Laboratory of Intelligent Technology and Systems

² Tsinghua National Laboratory for Information Science and Technology

³ Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China

maojiaxin@gmail.com, {yiqunliu, z-m, msp}@tsinghua.edu.cn

Abstract. Click-through information has been regarded as one of the most important signals for implicit relevance feedback in Web search engines. Because large variation exists in users' personal characteristics, such as search expertise, domain knowledge, and carefulness, different user clicks should not be treated as equally important. Different from most existing works that try to estimate the credibility of user clicks based on click-through or querying behavior, we propose to enrich the credibility estimation framework with mouse movement and eye-tracking information. In the proposed framework, the credibility of user clicks is evaluated with a number of metrics in which a user in the context of a certain search session is treated as a relevant document classifier. With an experimental search engine system that collects click-through, mouse movement, and eye movement data simultaneously, we find that credible user behaviors could be separated from non-credible ones with a number of interaction behavior features. Further experimental results indicate that relevance prediction performance could be improved with the proposed estimation framework.

Introduction

With the growth of information available on the Web, search engines have become a major information acquisition platform. Search engines try to retrieve and display *relevant* results with respect to the query issued by the user. To a large extent, the success of a Web search engine depends on how well it can estimate the relevance between a query-result pair. Recently, the interaction logs of users, especially the click-through data, have been shown to have great value in improving Web search engines. This is because they can be used as implicit relevance feedbacks from users and they are easy to collect on a large-scale (e.g. [11,12]). The most common approaches to exploit click-through data involve the training of *generative models* (named click models) to model users' click behaviors and extract relevance information (e.g. [3,2]).

However, not every click is a good indicator of relevance. During a search session, some users tend to rely on the snippets provided by search engines to determine the

* This work was supported by National Key Basic Research Program (2015CB358700) and Natural Science Foundation (61472206, 61073071) of China.

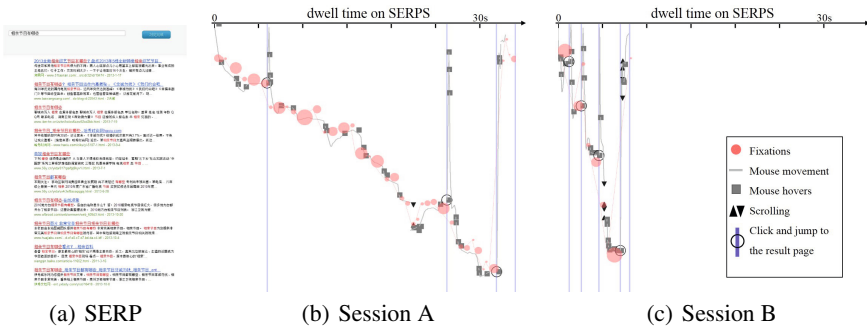


Fig. 1. Two search sessions of a same query. (a) shows the SERP of the query, (b) is for Session A and (c) is for Session B. In (b) and (c), we plot the y coordinates of the eye and mouse against the dwell time on the SERP. Note that the y coordinates are aligned across (a),(b) and (c). The size of the red circle indicates the duration of the fixation.

relevance of the results. They may read the SERPs (Search Engine Result Pages) and click on search results after thorough consideration. In such situations, these clicks are more likely to be informative indicators of users' relevance judgments. Other users may not be skilled at making relevance judgments or may click on multiple results in a short time without attentively selecting the results. In those case, the extent to which these clicks can be used as implicit relevance feedbacks is questionable.

Previous studies showed that search engine users have different behavioral patterns while browsing and clicking results on SERPs [21]. Users who use advance syntax [23] or possess domain knowledge [22] are regarded as *expert users* [7], and are expected to be more likely to provide credible relevance feedback information. These findings suggest that the clicks made by different users should be treated differently. Xing et al. [24] characterized the expertise of search engine users and proposed an unsupervised method to find expert users for improving relevance estimation. We follow the idea that we can regard the user as a relevant document classifier and then use some *metrics* to measure its performance. To take this analysis a step further, we claim that the credibility of clicks is not only related to the users' expertise in finishing search tasks but also depends on other factors, including the concentration and willingness of users. Different from previous works, which are usually based on merely click-through logs, we try to exploit a more comprehensive behavioral log, which involves eye-tracking and mouse movement behavior of search users. Rather than extracting *user-level* credibility estimation, we characterize click credibility on a *session-level*¹ because we believe that the relevance judgment performance of a certain user varies from query to query and should not be regarded as the same among different queries. Figure 1 shows two search sessions on a same query. The user in session A read the results in an approximate linear order, with the mouse following the gaze position, leaving many hovers on the results, and clicked on result 2, 7, 10. The user in session B browsed the SERP in a hurry, left fewer gazes and mouse hovers, and clicked on result 1, 3, 5, 9. Since the relevant results

¹ By *session* we mean a unique user-query pair in a continuous time period.

of this query are result 1, 2 and 7, we can see that click behaviors in Session A are more reliable because 2 of 3 clicks are on relevance results. This case serves as an example that the click credibility varies across sessions and user behavioral logs can be used to estimate it.

In this research, we try to determine the extent to which we can *trust* the click-through data by exploring more interaction behavior information collected from users. To investigate this problem, we built an experimental search engine with which we can collect detailed user behavioral data. The collected data include the click logs, mouse movement logs and eye-tracking information. We also hired accessors to manually annotate all the query-result pairs used in the experiment in 4-level relevance score. With this comprehensive dataset, we compared a variety of metrics to characterize the credibility of user clicks.

The major contributions of this paper can be summarized as follows:

- We proposed a framework and a variety of metrics to characterize the credibility of clicks on a session level.
- Based on a number of interaction behavior features extracted from mouse movement, eye movement and click-through behavior, we constructed a learning-based framework to estimate the credibility of search users' behaviors.
- We demonstrated that the predicted credibility of clicks can be used to improve the relevance estimation of the query-result pairs, which could help to improve the ranking performance of search engines.

Related Work

Click Models

Although click-through data are informative, previous works showed that they can not be interpreted as implicit relevance feedback directly because they are biased [12]. One of the major biases is the *position bias*. That is, the results that are ranked higher in the SERPs are more likely to be examined by users and therefore to be clicked by users. To address this bias, *click models* were proposed to estimate the probability of search result being examined and being clicked (e.g.[2,3]). In most of the click models, the *examination hypothesis* [3,16], that the user will click on a result *if and only if* he or she examines it and regards it as relevant, is made. Thus, after acquiring the examination information and the click-through data by mining the log, the relevance can be estimated.

Identifying Expert Users

According to previous works, the Web search engine users behave quite differently. [21] used a search trail extraction method to investigate the behavioral variability among users and among queries. They reported two extreme classes of users. The 'Navigators' have consistent interaction patterns, whereas the 'Explorers' have interaction patterns with a much higher variability. [23] compared the users who use advance syntax with other users. In addition to a significant difference in behavioral patterns, they found that the relevance scores of the results clicked by the advance syntax users were higher.

Similar research on users who are domain experts [22] also showed that these domain expert users are more likely to successfully find relevance results when issuing a in domain query.

Exploiting Eye-Tracking and Mouse Movement Information

Eye-tracking technology has attracted extensive attention from search engine researchers. Eye-tracking devices can record users' real-time eye movements and help to elucidate how users browse and interact with the SERP. [4] found that users spend more time examining the top results and that adding information to snippets will improve the performance for informational tasks. [1] found that search engine users have two evaluation styles. The Exhaustive style user will read more snippets before making a click, whereas the Economic style user only scans a few results before the first action.

Mouse movement information is another behavioral data source that worth investigating. Previous works [8,17] suggested that mouse positions and movements can be used to predict gaze positions. [6] exploited mouse movement data and click logs to predict the success of search sessions. Huang et al. found that *hovering* on results can be interpreted as implicit relevance feedback [10], and can be considered as a signal of examination in click models [9]. [18] built an end-to-end pipeline that can collect relevance annotations, click data and mouse movement data simultaneously. They also used several mouse movement features and the collected relevance annotations to train a classifier to predict the relevance of other results.

Our work is related to but different from these works. The click models enable us to estimate relevance by merely mining the click-through data. However, the click-through data are not always reliable, which motivates us to estimate the credibility of clicks. Additionally, the differences at the user level (e.g. the user's search expertise) is one of, but not the most, decisive influencing factors for the credibility. Thus in this paper, we leverage more interaction information, especially mouse movement data, to estimate click credibility at a session level, not a user level.

User Behavior Dataset

To analyze the session-level click credibility, we built an experimental search engine and recruited 31 subjects (15 males and 16 females) to conduct an experiment. The subjects were first-year university students from two different majors and with a variety of self-reported expertise in using search engines.

In the experiment, each subject was asked to accomplish 25 search tasks selected from NTCIR IMine task ². Each task corresponded to a specific query and necessary explanation to avoid ambiguity. We selected 5 navigational queries, 10 informational queries, and 10 transactional queries to cover various types of information needs. The queries and explanations are listed in Table 1. We crawled the first result page for each task from a Chinese commercial search engine and because the vertical results and the advertisement may affect users' behaviors [20], we filtered out the vertical results and advertisement. For each task, we restricted all the subjects to use the same query and browse the same SERP with 10 ordinary results.

² (<http://www.thuir.cn/imine>)

Table 1. Queries and explanations of 25 search tasks. For intent, 'I' indicates Informational, 'T' indicates Transactional, and 'N' indicates Navigational.

ID	Query	Translation	Explanation	Intent
1	央金兰泽的歌曲	Yangjinlanze's song	Find the information of the music sung by the singer	I
2	卫子夫	Empress Wei Zifu	Find the introduction of the historical figure	I
3	天梭手表官网	Official website of Tissot	Find the official website of a watch brand	N
4	温柔的谎言	Gentle lies	Find the online watching resources of a TV drama	T
5	佛教音乐	Buddhist music	Find Buddhist music download resources	T
6	声卡是什么	What is a sound card	Find the definition and principal of sound cards	I
7	qq加速器下载	qq accelerator download	Find the download resources of a software	T
8	浏览器下载	Browser download	Find the download resources of Web browsers	T
9	相亲节目有哪些	What dating shows are there on TV	Find major TV dating shows	I
10	遮天	Zhe Tian	Find online reading resources of a novel	T
11	工行网上银行个人网上银行	Personal online bank of CCBC	Find the website of a bank	N
12	冬季恋歌国语全集	Chinese collections of Winter Sonata	Find the online watching resources of a TV drama	T
13	乘法口诀	Multiplication table	Find the multiplication table for pupils	I
14	学雷锋作文	Writings of 'learn from Lei Feng'	Find writing samples of a given topic for pupils	I
15	驾驶证考试网	Website of driving license exam	Find the website for the driving license exam	N
16	春雨的诗句	Poems of spring rain	Find famous poems for spring rain	I
17	初恋这件小事	First Love	Find online watching resources for a movie	T
18	魅族官网	Official website of Meizu	Find the official website of a digital brand	N
19	斯特拉马乔尼	Stramaccioni	Find the resume of a soccer coach	I
20	哈利波特	Harry Potter	Find online watching resources for a movie	T
21	电脑桌面壁纸	Wallpaper for computer	Find wallpaper pictures for a desktop computer	T
22	清明上河园	Millennium City Park	Find photos of a viewpoint	I
23	安卓2.3游戏下载	Android 2.3 game download	Find download resources of mobile games	T
24	陈楚生演唱会	Chen Chusheng's concert	Find the concert information of a singer	I
25	联通网上营业厅	Online service of China Unicom	Find the official website of a mobile company	N

To analyze each user's examination processes, we used a Tobii X2-30 eye-tracker to record each subject's eye movements during the experiment. The data recorded by the eye-tracker are sequences of fixations and saccades (fast eye movements between two fixations). According to [13], when reading, the subject will process only the content that he or she fixates on. Therefore, to reconstruct the examination sequence, we only took the fixations into account and regarded the fixation on a certain search result as a signal that the subject had examined it.

A variety of mouse events, including mouse movements, hovers on results, scrolling and clicks on result, were logged by injecting Javascript code into the crawled SERPs. We also hired 3 annotators to give objective relevance judgments of all the results. The relevance score had 4 levels, with 1 for least relevant and 4 for most relevant. We regarded results with a relevance score of 3 or 4 as relevant results, and the result with a score of 1 or 2 as not relevant. When disagreement occurred, we used the *median* of the scores from the three annotators as the final score.

From these 31 subjects, we collected 774 valid query sessions (one of the query sessions was abandoned because the eye-tracker malfunctioned during the experiment). Each query session has a comprehensive behavioral log and can be uniquely identified by the $\langle user, query \rangle$ pair.

Estimating Session-level Credibility

Credibility Metrics

When using a Web search engine, users always aim to find information or to accomplish a certain task, related to the issued query. During a search session, the user will obtain some information from the SERP and then click on the result that he or she believes to be helpful in fulfilling the information need. We can view this as a classification task [24]. The user is a classifier that takes the results as samples, the obtained information as input features, and outputs corresponding relevance judgments, by performing clicks or skips (no clicking after examination) over the results. Therefore, to use click-through data as implicit relevance feedbacks is to aggregate the outputs of multiple classifiers. The credibility of the click-through data generated in the session *is* the credibility of the output of the classifier, given a specific SERP as the input features. This analogy inspires us to use metrics that measures the performance of a binary classifier to characterize the session-level click credibility.

The confusion matrix [19] is a common evaluation metric in classification. We regard relevant results as positive samples and irrelevant results as negative samples. Each row of the matrix represents the instances in one of the predicted classes, which in our case is whether the user click or skip on the result, whereas each column represents the instances in one of the actual classes, that is, whether the result is actually relevant to the query. We assume that a user's click credibility is independent of the content (the results on SERPs) that he or she did not examine. This assumption is necessary because a user may not always examine all the results on the SERPs, and it does not make sense to judge the user's click credibility according to the results he or she did not examine. We used eye-tracking data as an explicit information source for examination to filter out the unexamined results. Based on the examination sequence, the click log of a query

session and the relevance score given by the annotators, we can compute the confusion matrix of the session. From the matrix, we derived 3 intuitive metrics for session-level click credibility:

- *Accuracy*: $\frac{\#\{TP\}+\#\{TN\}}{\#\{TP\}+\#\{TN\}+\#\{FP\}+\#\{FN\}}$ ³. The proportion of correctly classified samples.
- *True Positive Rate*: $\frac{\#\{TP\}}{\#\{TP\}+\#\{FN\}}$. The proportion that a relevant result is clicked by the user. True positive rate is also referred to as recall, and related to the user’s ability to find relevant result on SERPs.
- *True Negative Rate*: $\frac{\#\{TN\}}{\#\{TN\}+\#\{FP\}}$. The proportion that an irrelevant result is skipped by the user. It relates to the user’s ability to prevent unnecessary clicks, and it is called specificity in some situations.

The statistics on the metrics are showed in Table 2. For each of the metrics, we group the $\langle user, query \rangle$ pairs by the users and by the queries, separately compute the means and standard deviations of each group and show the macro average mean M and macro average standard deviation SD. We noticed obvious variability in all 3 metrics, which indicates the credibility of clicks varies across sessions. The standard deviations of the accuracy and true positive rate are lower when grouped by query, which reveals that the query is a more determining factor than the user. Therefore, it is necessary to estimate the credibility of clicks on the session level, not the user level.

Table 2. Statistics for metrics of click credibility

Metrics		Accuracy	True Positive Rate	True Negative Rate
Grouped by user	<u>M</u>	0.613	0.557	0.813
	<u>SD</u>	0.236	0.274	0.239
Grouped by query	<u>M</u>	0.613	0.557	0.811
	<u>SD</u>	0.200	0.257	0.247
Single session	<u>M</u>	0.613	0.557	0.812
	<u>SD</u>	0.249	0.306	0.269

Predicting Click Credibility

We have proposed 3 metrics to characterize the session-level click credibility. However, our primary concern is to find credible clicks to improve automated relevance feedback and the computation of the proposed metrics themselves requires manually annotation of the relevance. Therefore, we need to *predict* the metrics of click credibility when the relevance annotation is not available. In this section, we try to use the behavioral logs as features and predict the click credibility through a *regression* process. Because all the proposed metrics are probabilities $p \in [0, 1]$, it is more convenient to predict the corresponding *log-odds* $\alpha \in (-\infty, +\infty)$ of them:

$$\alpha = \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

³ Here $\#\{TP\}$, $\#\{TN\}$, $\#\{FP\}$ and $\#\{FN\}$ are used to denote the number of true positives, true negatives, false positives and false negatives.

Feature Extraction. To make large scale deployment possible, we only used data that can be collected in practical application scenarios of search engines. That is, we did not use eye-tracking data and relevance annotations to generate the features. The features that we used and their correlations (measured in Pearson’s r) with the metrics are list in Table 3. The click features are the features that can be generated from click-through logs. The session features are some time-related features of a certain session. The mouse movement features include hovers, mouse movement and scrolling information. Feature 10, the unclicked hovers, were proposed in [10]. The click entropy [5] of a query is defined as:

$$\text{ClickEntropy}(q) = - \sum_i P(C_i = 1|q) \log(P(C_i = 1|q))$$

The N Results Satisfied Rate [14] is the proportion of the sessions in which only first N results were clicked. For the user features, we computed the click entropies of users, which is similar to the click entropies of queries, and also includes the total click numbers and time-related features.

Evaluations. The ground truths of proposed metrics are calculated using the eye-tracking data and the click-through data. As the whole dataset was limited in size (774 samples), we use the Leave-One-Out Cross Validation (LOO-CV) to access the generalized prediction performance. Because Web search engines possess a large number of users and response to all kinds of queries, it is reasonable to assume that when predicting the session-level click credibility, the user and the query are *unseen* in the training set. Therefore, in every iteration, we chose one query and one user, the corresponding $\langle \text{user}, \text{query} \rangle$ pair was then used as test set, and the $\langle \text{user}, \text{query} \rangle$ pairs of the other 24 queries and the other 30 users formed the training set.

A series state-of-the-art regression methods, including the SVR, random forest regression, gradient boosting tree regression and Lasso⁴, were tested using the training data. We finally used SVR, which is relatively stable and gains promising results, to predict the click credibility.

If behavioral information is not available, the best estimate of each of the metrics is the arithmetic mean of the metrics of the training set sessions. We used it as the baseline estimation method because to our best knowledge, there are no existing works which try to estimate user behavior credibility with mouse movement information. The results, measured in Mean Squared Error (MSE) and Mean Absolute Error (MAE) with the ground truth, are listed in Table 4.

Estimating Relevance

Having obtained the predicted log-odds α , we can compute the corresponding predicted metrics by $p = e^\alpha / (1 + e^\alpha)$, and use them to improve the relevance estimation. Recalling the *examination hypothesis* [3], we can estimate the relevance score r_i by:

$$r_i = P(C_i = 1 | E_i = 1) \tag{1}$$

⁴ Implementations are provided in scikit-learn [15].

Table 3. Features extracted from the behavioral log with Pearson’s r between features and metrics (TP: True Positive Rate, TN: True Negative Rate). * indicates $r \neq 0$ with $p < 10^{-3}$.

No.	Description	Accuracy	TP	TN
Click Features				
1	number of clicked results	-0.02	0.24*	-0.55*
2	lowest rank of clicks	-0.11	0.05	-0.36*
3	average difference in ranks between two clicks	-0.06	0.03	-0.16*
Session Features				
4	average time spent on the SERP for each click	-0.04	-0.05	0.09
5	total time spent on the search task	-0.07	0.03	-0.19*
6	total time spent on the SERP	-0.12*	-0.01	-0.22*
7	maximal continuous time spent on the SERP	-0.14*	-0.16*	0.06
Mouse Movement Features				
8	number of hovered results	-0.15*	-0.06	-0.26*
9	lowest rank of hovers	-0.18*	-0.10	-0.24*
10	number of results that are hovered over but not clicked	-0.18*	-0.23*	0.07
11	moving time of the mouse	-0.12	-0.02	-0.17*
12	idle time of the mouse	-0.06	0.10	-0.34*
13	dwell time of the mouse in the result region	-0.11	-0.03	-0.16*
14	length of the mouse trails	-0.08	0.08	-0.34*
15	velocity of mouse movement	0.12*	0.17*	-0.16*
16	horizontal moving distance of the mouse	-0.05	0.07	-0.19*
17	vertical moving distance of the mouse	-0.08	0.08	-0.37*
18	maximal y coordinate that the mouse has reached	-0.18*	-0.11	-0.22*
19	total distance of scrolling	-0.11	-0.09	-0.13
20	maximal displacement in y axis of scrolling	-0.15*	-0.13*	-0.14*
Query Features				
21	click entropy of the query	-0.17*	-0.13*	-0.13
22	N Results Satisfied Rate of the query	-0.08	-0.03	0.01
User Features				
23	click entropy of the user	-0.16*	-0.06	-0.21*
24	user’s total number of clicks	-0.02	0.14*	-0.32*
25	average time that the user spends on each search task	-0.08	0.02	-0.20*
26	average time that the user spends on the SERP	-0.10	-0.05	-0.12

Table 4. Results of predicting click credibility, * indicates the improvement over baseline is significant with $p < 10^{-3}$

	Baseline		SVR	
	MSE	MAE	MSE	MAE
Accuracy	0.071075	0.224086	0.061015(-14.1%*)	0.206813(-7.7%*)
True Positive Rate	0.109941	0.290940	0.082651(-24.8%*)	0.219068(-24.7%*)
True Negative Rate	0.086662	0.192934	0.069311(-20.0%*)	0.165896(-16.6%*)

Where $E_i = 1$ indicates that result i is examined and $C_i = 1$ indicates that result i is clicked. If we know the accuracy of the session, a_s , then we have:

$$\begin{aligned} P(C_i = 1|E_i = 1) &= r_i \times a_s + (1 - r_i) \times (1 - a_s) \\ P(C_i = 0|E_i = 1) &= (1 - r_i) \times a_s + r_i \times (1 - a_s) \end{aligned} \quad (2)$$

If we know the true positive rate TP_s and the true negative rate TN_s , we have:

$$\begin{aligned} P(C_i = 1|E_i = 1) &= r_i \times TP_s + (1 - r_i) \times (1 - TN_s) \\ P(C_i = 0|E_i = 1) &= (1 - r_i) \times TN_s + r_i \times (1 - TP_s) \end{aligned} \quad (3)$$

By the criterion of maximum likelihood estimation, we can estimate r_i . We use the fixation information recorded by eye-tracker as explicit signal to estimate E_i . That is, during a session, if the user fixates on result i , we assume that he or she has examined this result and set $E_i = 1$. We use the relevance score given by (1) as the baseline, and refer to (2) as the accuracy model, (3) as the confusion matrix model.

To evaluate the performance in estimating the relevance, we use the predicted relevance score r_i to re-rank the results on SERP in a descending order, and use the Mean Average Precision (MAP) to measure the ranking performance. Higher MAP is associated with better estimation performance. We use the metrics computed by LOO-CV as the predicted metrics. The results are in Table 5. Both models can improve original rankings and significantly outperform the baseline; therefore, the predicted metrics can be utilized to improve relevance estimation. The accuracy model is slightly better than the confusion matrix model.

Table 5. Results of relevance estimation, * indicates the improvement is significant with $p < 0.05$ and ** indicates $p < 0.01$

	Original Ranking	Baseline (Examination hypothesis without credibility estimation)	Accuracy Model	Confusion Matrix Model
MAP	0.805124	0.843075	0.884604	0.877654
Improvement over original ranking	-	+4.7%	+9.9%**	+9.0%**
Improvement over baseline	-	-	+4.9%**	+4.1%*

Conclusions and Future Works

To estimate the credibility of click-through data of Web search engines, we conducted a user behavioral experiment to analyzed a detailed behavioral log. Based on the analogy that a search engine user can be regarded as a classifier that classifies the search results into a relevant class or an irrelevant class, we used 3 metrics, i.e, the accuracy, the true positive rate and the true negative rate, which are designed for measuring classification

performance, to measure session-level click credibility. We reported the correlations between these click credibility metrics and user behavioral features, and demonstrated that we can use the predicted metrics to improve implicit relevance feedback of search engines.

These metrics not only provide us with extra dimensions in understanding the interactions between users and search engines, but can be also applied to improving search engines. Therefore, our work can be considered as a demonstration of a framework that exploits the abundant but complex behavioral data. That is, we can extract some properties (e.g. the proposed click credibility metrics), which are either intuitive or useful, from the query sessions and try to predict them using the behavioral data and a supervised learning method. This process is a user behavior analysis that may produce new insights. And at the same time, we can use these properties to improve the search engine itself.

In future works, we will further explore more behavior-related properties, which may either help the user behavioral analysis or the enhancement of other search engine functionalities. On the other direction, we will try to collect mouse movement logs on a larger scale and validate our approach in a real-world dataset.

References

1. Aula, A., Majaranta, P., Rähkä, K.-J.: Eye-tracking reveals the personal styles for search result evaluation. In: Costabile, M.F., Paternó, F. (eds.) *INTERACT 2005*. LNCS, vol. 3585, pp. 1058–1061. Springer, Heidelberg (2005)
2. Chapelle, O., Zhang, Y.: A dynamic bayesian network click model for web search ranking. In: *Proceedings of the 18th International Conference on World Wide Web*, pp. 1–10. ACM (2009)
3. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 87–94. ACM (2008)
4. Cutrell, E., Guan, Z.: What are you looking for?: an eye-tracking study of information usage in web search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 407–416. ACM (2007)
5. Dou, Z., Song, R., Wen, J.R.: A large-scale evaluation and analysis of personalized search strategies. In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 581–590. ACM (2007)
6. Guo, Q., Lagun, D., Agichtein, E.: Predicting web search success with fine-grained interaction data. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 2050–2054. ACM (2012)
7. Hölscher, C., Strube, G.: Web search behavior of internet experts and newbies. *Computer Networks* 33(1), 337–346 (2000)
8. Huang, J., White, R., Buscher, G.: User see, user point: gaze and cursor alignment in web search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1341–1350. ACM (2012)
9. Huang, J., White, R.W., Buscher, G., Wang, K.: Improving searcher models using mouse cursor activity. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 195–204. ACM (2012)
10. Huang, J., White, R.W., Dumais, S.: No clicks, no problem: using cursor movements to understand and improve search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1225–1234. ACM (2011)

11. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)
12. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting click-through data as implicit feedback. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 154–161. ACM (2005)
13. Just, M.A., Carpenter, P.A.: A theory of reading: From eye fixations to comprehension. *Psychological Review* 87, 329–354 (1980)
14. Liu, Y., Zhang, M., Ru, L., Ma, S.: Automatic query type identification based on click through information. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) AIRS 2006. LNCS, vol. 4182, pp. 593–600. Springer, Heidelberg (2006)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
16. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: Estimating the click-through rate for new ads. In: Proceedings of the 16th International Conference on World Wide Web, pp. 521–530. ACM (2007)
17. Rodden, K., Fu, X., Aula, A., Spiro, I.: Eye-mouse coordination patterns on web search results pages. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 2997–3002. ACM (2008)
18. Speicher, M., Both, A., Gaedke, M.: Tellmyrelevance!: Predicting the relevance of web search results from cursor interactions. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 1281–1290. ACM (2013)
19. Stehman, S.V.: Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* 62(1), 77–89 (1997)
20. Wang, C., Liu, Y., Zhang, M., Ma, S., Zheng, M., Qian, J., Zhang, K.: Incorporating vertical results into search click models. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2013, pp. 503–512. ACM, New York (2013), <http://doi.acm.org/10.1145/2484028.2484036>
21. White, R.W., Drucker, S.M.: Investigating behavioral variability in web search. In: Proceedings of the 16th International Conference on World Wide Web, pp. 21–30. ACM (2007)
22. White, R.W., Dumais, S.T., Teevan, J.: Characterizing the influence of domain expertise on web search behavior. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 132–141. ACM (2009)
23. White, R.W., Morris, D.: Investigating the querying and browsing behavior of advanced search engine users. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 255–262. ACM (2007)
24. Xing, Q., Liu, Y., Zhang, M., Ma, S., Zhang, K.: Characterizing expertise of search engine users. In: Banchs, R.E., Silvestri, F., Liu, T.-Y., Zhang, M., Gao, S., Lang, J. (eds.) AIRS 2013. LNCS, vol. 8281, pp. 380–391. Springer, Heidelberg (2013)

Cannabis_TREATS_cancer: Incorporating Fine-Grained Ontological Relations in Medical Document Ranking

Yunqing Xia¹, Zhongda Xie¹, Qiuge Zhang², Huiyuan Wang², and Huan Zhao³

¹ Department of Computer Science, TNList,
Tsinghua University, Beijing 100084, China
{yqxia,xzd13}@tsinghua.edu.cn

² Information Networking Institute
Carnegie Mellon University, Pittsburgh, PA 15213, USA
{zqg0830,wanghuiyuan.anna}@gmail.com

³ Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
hzhaoaf@ust.hk

Abstract. The previous work has justified the assumption that document ranking can be improved by further considering the coarse-grained relations in various linguistic levels (e.g., lexical, syntactical and semantic). To the best of our knowledge, little work is reported to incorporate the fine-grained ontological relations (e.g., $\langle \textit{cannabis}, \textit{TREATS}, \textit{cancer} \rangle$) in document ranking. Two contributions are worth noting in this work. First, three major combination models (i.e., summation, multiplication, and amplification) are designed to re-calculate the query-document relevance score considering both the term-level Okapi BM25 relevance score and the relation-level relevance score. Second, a vector-based scoring algorithm is proposed to calculate the relation-level relevance score. A few experiments on medical document ranking with CLEF2013 eHealth Lab medical information retrieval dataset show that the proposed document ranking algorithms can be further improved by incorporating the fine-grained ontological relations.

Keywords: Medical document ranking, ontological relation, medical concept, relevance.

1 Introduction

Nowadays, powerful search engines are available and people may consult the search engines with queries like *cannabis and cancer*. The underlying information need is actually the connection between the two things rather than information of the two things. Considering the following three sentences:

(S#1): He suffers from *cancer* but he never quits *cannabis*.

(S#2): Studies prove that *cannabis* can be an effective treatment for

cancer.

(S#3): The report indicates that long-term *cannabis* use may cause lung *cancer*.

Now, let's use *cannabis and cancer* as the query. Considering merely terms, we would find that the three sentences are equally relevant to the query. However, sentence S#1 is not truly relevant because no relation occurs between term *cannabis* and *cancer* though both terms are mentioned. Such a mistake occurs in medical information retrieval systems because ontological relation is not considered in document ranking. In our research in medical information retrieval, we find more than 20 percent queries usually involve fine-grained ontological relations, e.g., $\langle drug, TREATS, disease \rangle$.

The previous work has justified the assumption that relations of various linguistic levels are helpful to improve document ranking [1–9]. A majority of research is conducted on statistical term dependency. Other work is conducted on syntactic dependency and semantic relation. Undoubtedly, the above coarse-grained relations are useful, but the discovered relations are lack of meaning. For example, $\langle thingX, ISA, thingY \rangle$ indicates a general hypernymous relation, in which *thingX* and *thingY* can be any things.

In medical domain, an early work is reported in [3] which made use of fine-grained ontological relations between medical concepts in cross-language medical information retrieval. Enlightened by the positive results, we conduct a further study which handles the fine-grained medical documents and applies them to enhance medical document ranking.

In this work, we design three combination models (i.e., summation, multiplication and amplification) to calculate the new query-document relevance score, which combines the term-level relevance score and the relation-level relevance score. We calculate the term-level relevance score with the standard Okapi BM25 algorithm¹ in Lucene². For the relation-level relevance, we design a vector-based scoring algorithm which first represents query and documents with eighteen-dimension vectors, and then calculates the relevance score using cosine formula. A few experiments are conducted in medical document ranking task with dataset from CLEF2013 eHealth Lab on medical information retrieval, which show that the proposed document ranking algorithms can be further improved by incorporating the fine-grained ontological relations.

The reminder of this paper is organized as follows. In Section 2, we summarize the related work. In Section 3, we present our document ranking method. We present evaluation as well as discussion in Section 4 and conclude this paper in Section 5.

2 Related Work

The early attempts to incorporate relations in textual information retrieval (IR) started are based on concepts or semantics. A concept-based solution is

¹ http://en.wikipedia.org/wiki/Okapi_BM25

² Lucene: <http://lucene.apache.org/>

discussed in [10], in which term dependencies are first taken into account. Later on, lexical-semantic relations are used in [11] to build a structured representation of documents and queries. Due to shortage of large-scale semantic knowledge resource, the semantics based approach improves IR system slightly. Instead, Khoo et al. (2001) focused on merely cause-effect relations [12]. In [13], a relation-based search engine, i.e. OntoLook, constructs a concept-relation graph to model semantic relationships amongst concepts. In [14], semantic relationships are revisited using ontology. We notice that the semantic relations used in these IR systems are coarse-grained. That is, most relations formalize general connections of things but have little to do with real meaning.

In late 1990's, researchers started to study effect of syntactic term dependency relation on information retrieval. Syntactic term dependency was first used in [2] to improve Japanese information retrieval. However, the syntactic parsing tools at that moment were slow and less accurate. Recently, Park et al. (2011) proposed a quasi-synchronous dependence model based on syntactic dependency parsing for both queries and documents [7].

In the meantime, a majority of research is conducted to incorporate statistical term dependency in IR system. A general language model was proposed in [1] which presents word dependency with bi-grams. Gao et al. (2004) proposed a dependency language model in ranking documents based on statistical term dependency (i.e., linkage) [4]. This model is later revised in [6] with more general term dependency in syntactic and semantic levels. Very recently, statistical high-order word association relation was exploited by [8] in document ranking using pure high-order dependence among a number of words. Term association was further studied in [9] for probabilistic IR by introducing a new concept cross term in modeling term proximity.

It should be noted that relations have been also used in question answering. In [15], a general rank-learning framework was proposed for passage ranking within question answering systems using linguistic and semantic features. Semantic relations were also discussed in [5] to improve question analysis and answer retrieval.

Little research work is conducted to incorporate relations in medical IR since Vintar et al. (2003) achieved positive results with the fine-grained ontological relations [3], in which the relations are used to filter cross-lingual web pages in a boolean manner. In the past five years, TREC medical track [16] and CLEF eHealth Lab [17] were organized to advance the research on medical IR. However, no medical IR system uses fine-grained ontological relations in document ranking. Start from Vintar et al. (2003)[3], we design a unified ranking algorithm which combines the traditional BM25 relevance score and the proposed relation-level relevance score. The difference lies in that the relevance score is re-calculated in our work.

3 Methodology

The core of this work is assigning each document a refined relevance score that reflects relevance of a document to the query considering not only terms but also

fine-grained ontological relations. This goal is achieved in three steps. First, we calculate the term-level relevance score using BM25. Second, we calculate the relation-level relevance score using our method. At last, we adopt the following three popular combination models to calculate the refined relevance score (r^*) by combing the term-level relevance score (r) and the proposed relation-level relevance score (l):

- Summation

$$r^* = \alpha \times r + (1 - \alpha) \times l \quad (1)$$

in which α is the normalization factor.

- Multiplication

$$r^* = r \times l \quad (2)$$

- Amplification

$$r^* = r \times \beta^l \quad (3)$$

in which β is the exponential base. We set $\beta = e$ in our study according to empirical study.

This section focuses on calculation of the relation-level relevance score, i.e. l_p , which is achieved as follows: First, ontological relations are discovered within text of query and documents; Second, query and documents are represented with relation vector; Third, relational relevance score is calculated by comparing the vectors; In what follows, we elaborate the key modules in our document ranking method.

3.1 Ontological Relation Discovery

In Wikipedia, *ontology* is defined as *the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations*³. According to this definition, we further define the ontological relations as follows.

Definition: *Ontological relation*

An ontological relation is defined as the real-world relation between existential beings (things or events).

Compared with the general semantics relations such as synonym and polysemy, the ontological relations reflect fine-grained real-world semantic relationship such as *person_PRESIDENT_OF_nation* and *medicine_CURES_disease*.

For the three example sentences in Section 1, the corresponding ontological relations are given below:

³ <http://en.wikipedia.org/wiki/Ontology>

- (R#1): NULL
 (R#2): *cannabis_TREATS_cancer*
 (R#3): *cannabis_CAUSES_cancer*

Compiling the ontological relations is a tricky job, even for the specific medical domain. Fortunately, 57 types of ontological relations are defined in SemMedDB [18]. However, some relations are either overlapping with other relations or less important according to statistics in SemMedDB. To simplify the problem, we employ three medical experts to handle relations manually. Finally, an agreement on the following eighteen relations is reached: PROCESS_OF, METHOD_OF, LOCATION_OF, PART_OF, OCCURS_IN, STIMULATES, MANIFESTATION_OF, CONVERT_TO, AUGMENTS, ASSOCIATED_WITH, PREVENTS, USES, TREATS, PREDISPOSES, PRODUCES, DISRUPTS, CAUSES and INHIBITS.

To be formal, the ontological relation is represented by a three-tuple: $\langle C\#1, r, C\#2 \rangle$, where $C\#1$ and $C\#2$ represent two medical concepts and r a relation. The medical concepts are obtained with MetaMap⁴.

In this work, ontological relation is discovered based on keywords which are mentioned in SemMedDB annotations of predicate instances. To reduce complexity, we only use the high-frequency ones (i.e., 8,015 unigram keywords and 114,839 bigram keywords). We find some keywords indicate different relation in different context. Thus the discovered ontological relations can be modeling in a probabilistic manner with a priori distribution within texts. We thus extend the above four-tuple to include relation probabilities to 20-tuple: $\langle C\#1, \{r_1; p_1\}, \dots, \{r_{18}; p_{18}\}, C\#2 \rangle$, where r_i represents the i -th relation and p_i its probability, which is estimated in SemMedDB using the simple MLE (maximum likelihood estimation) technique.

3.2 Representation of Query and Document Using Ontological Relations

Considering an 18-dimension vector that entails the aforementioned 18 relations, we now create a relation vector $V = \{r_1 : w_1, \dots, r_{18} : w_{18}\}$ for a piece of medical text. We use relation keywords mentioned in Section 3.1 in detecting ontological relations in text. We map query and document to relation vectors with different approaches.

(1) Query

Query is usually too short to indicate a deterministic medical relation, especially when no keyword is mentioned. We choose to assign equal probability to each possible relation. We first consult the UMLS⁵ and extract all possible relations that may occur between concepts via the keyword (if any) in the query. Then

⁴ MetaMap is a medical concept annotation tool available at <http://mmtx.nlm.nih.gov/>

⁵ UMLS is a medical knowledge base that can be downloaded via <http://www.nlm.nih.gov/research/umls/>.

equal probability is assigned to the relations and we obtain a relation vector for the query. For the query *cannabis and cancer*, we obtain a relation vector below:

$$(RV\#0) (0,0,0,0,0.5,0,0,0,0.5,0,0,0,0,0,0,0,0,0)$$

where the 5-th relation is TREATS and 9-th relation is CAUSES.

(2) Document

Document may mention the query for more than one times. Thus we need to resolve medical relation for each mention. For presentation convenience, we need first to fix the window for relation detection. Here, we use sentence as an example window in the following description. We map each query-mentioning sentence to a relation vector.

Difference between sentence and query lies in that sentence gives a much larger context thus can indicate a deterministic relation of a keyword. Thus the relation vector for a sentence contains only one non-zero value dimension. For the three example sentences in Section 1, we obtain three sentence-level relation vectors (RV) below:

$$\begin{aligned} (RV\#1) & (0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0) \\ (RV\#2) & (0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0) \\ (RV\#3) & (0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0) \end{aligned}$$

After relation vector for each window is obtained, we merge all these vectors thus obtain an overall vector for the document.

3.3 Calculating the Relation-Level Relevance Score

Once query and documents are represented with 18-dimension vectors, we are able to adopt vector-based distance measures in relation scoring. Given two vectors $v_i = \{w_{i1}, w_{i2}, \dots, w_{i18}\}$ and $v_j = \{w_{j1}, w_{j2}, \dots, w_{j18}\}$, we adopt the Cosine distance measure in distance calculation:

$$Cos(v_i, v_j) = \frac{\sum_{k=1}^{18} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{18} w_{ik}^2 + \sum_{k=1}^{18} w_{jk}^2}} \quad (4)$$

4 Evaluation

Data

We used the dataset in CLEF2013 eHealth Lab Medical IR task [17] in our experiments, which covers a broad range of health topics, targeted at both the general public and healthcare professionals.

The test queries are extracted from the 50 queries in CLEF2013 Medical IR task. As we intend to prove contribution of medical relations to medical document ranking, we select queries that involves more than one medical concepts.

Finally, we obtain fourteen queries: #6, #7, #8, #11, #12, #16, #17, #18, #23, #24, #25, #39, #40 and #49.

The CLEF2013 Medical IR dataset, denoted with CLEF, contains 1,878 relevant documents judged by nurses from the pool of 6,391 documents. As NA (non-annotation) documents are retrieved by our method, we employed three medical students to assess relevance of these documents. We calculate Kappa coefficient value between every two assessors and obtain the average Kappa coefficient value 0.82. In this way, we obtained an extended dataset, denoted with CLEF+.

Evaluation Metrics

Two metrics are used in our evaluation: (1) p@10: precision at top 10 web pages. (2) nDCG@10: normalized Discounted Cumulative Gain at the top 10 returned web pages. (3) MAP: Mean average precision at top 10 returned web pages.

4.1 Experiment 1: Methods

Three methods are compared in this evaluation, three of which are different implementations of our method:

- **BM25**: Okapi BM25 is used in relevance scoring. Default settings are adopted in the BM25 algorithm.
- **BMB**: The method described in [3] is implemented, in which the discovered ontological relations are used to filter the web pages in a *boolean* manner.
- **BMR**: Our method is implemented to combine BM25 term-level relevance score and relation-level relevance score. In this experiment, we adopt the the amplification formula as the combination model (Eq.3 and HTML tag pair as relation detection window. Such a setting is proved most effective in our experiments.

Experimental results are given in Table 1.

Table 1. Results of the document ranking methods

Method	p@10		nDCG@10		MAP@10	
	CLEF	CLEF+	CLEF	CLEF+	CLEF	CLEF+
BM25	0.450	0.516	0.448	0.504	0.112	0.129
BMB	0.437	0.521	0.435	0.514	0.106	0.117
BMR	0.456	0.534	0.452	0.519	0.124	0.144

It can be seen in Table 1 that when ontological relations are incorporated, the BMB method performs slightly worse than the BM25 baseline. This indicates that the boolean combination method does not bring performance gain. As a comparison, the proposed BMR method outperforms BM25 by 0.018 on p@10, by 0.015 on MAP@10 and by 0.015 on nDCG@10. Looking into the fourteen

queries which involve ontological relations, we find our proposed method obtains improvement on 9 queries while loss on 2 queries. The major reason for the loss is that some discovered relations in web pages are incorrect. This reminds us to plan future work on a better relation detection method.

Note that on CLEF+ dataset, BMR method improves more than that on CLEF dataset. This is because a few out-of-pool web pages are judged relevant by annotators in our work.

4.2 Experiment 2: Combination Models

In this experiment, we seek to compare the combination models described in Section 3. Accordingly, the following implementations of our method is evaluated:

- **SUMM**: Our method is implemented to adopt Eq.1 in combining the BM25 term-level relevance score and relation-level relevance score. We set $\alpha = 0.7$ in this implementation according to empirical study.
- **MULT**: We adopt Eq.2 in this implementation of our Our method.
- **AMPL**: We adopt Eq.3 in this implementation of our Our method.

Note in all the three BMR implementations, the relation detection window is set HTML tag pair, which is proved effective in our experiments. Results are presented in Table 2.

Table 2. Results of the our methods using different combination models

Method	p@10		nDCG@10		MAP@10	
	CLEF	CLEF+	CLEF	CLEF+	CLEF	CLEF+
SUMM	0.451	0.527	0.447	0.511	0.121	0.140
MULT	0.447	0.521	0.443	0.501	0.117	0.139
AMPL	0.456	0.534	0.452	0.519	0.124	0.144

We can see in Table 2 that the AMPL implementation performs best on both datasets across the three evaluation metrics. This justifies that the amplification model is advantageous over the two models. Looking into the output results, we find the MULT implementation improves BM25 method on 7 queries while loss on 3 queries.

4.3 Experiment 3: Relation Detection Window

This experiment aims to compare different relation detection windows in the proposed BMR method. The following six implementations are developed:

- **CURS**: The current sentence is used as relation detection window.
- **CURSP**: The current and the preceding sentence are used as relation detection window.

- **CURSPF**: The current and the preceding and following sentences are used as relation detection window.
- **CURP**: The current paragraph is used as relation detection window.
- **CURD**: The current web document is used as relation detection window.
- **HTML**: Text in the current HTML tag pair (e.g., TD and UL) is used as relation detection window.

Table 3 presents the experimental results of our method with the six different windows.

Table 3. Results of the our method with different relation detection window

Method	p@10		nDCG@10		MAP@10	
	CLEF	CLEF+	CLEF	CLEF+	CLEF	CLEF+
CURS	0.450	0.522	0.445	0.511	0.110	0.131
CURSP	0.451	0.524	0.448	0.513	0.111	0.134
CURSPF	0.453	0.528	0.449	0.516	0.112	0.137
CURP	0.442	0.476	0.431	0.502	0.106	0.127
CURD	0.427	0.458	0.416	0.489	0.097	0.112
HTML	0.456	0.534	0.452	0.519	0.124	0.143

Seen in Table 3 that the best window for relation detection is HTML. When the window is extended to the whole document, our method becomes worse than the BM25 baseline. This is because more errors occur in detecting relations in the whole document. Using paragraph as window also brings some errors. On the other hand, when the window is reduced to current sentence, quality of our method drops most. This is because in a sentence, many relations cannot be detected. However, the elements of these ontological relation can be found in the preceding or following sentences. This is why CURSPF outperforms CURSP and CURS. Meanwhile, a bigger context may bring noise. Thus the appropriate window is found HTML tag. This ascribes the writing style in HTML web pages.

5 Conclusion and Future Work

In this work, we propose a novel medical document ranking method which incorporates the fine-grained ontological relations in relevance scoring. The relation-level relevance score is measured by comparing relation vectors for query and documents. In our experiments, we evaluate not only the outperformance of our method over the state-of-the-art baseline methods, but also the influence of combination model and relation detection window on our method. Experimental results confirm that the ontological relations indeed bring performance gain in medical document ranking.

However, this work is still preliminary. For example, the eighteen types of ontological medical relations are compiled by human experts. We will explore the possibility to extend these relations to cover all possible medical relations.

The future work includes a finer algorithm for medical relation detection, a probabilistic model for relation-level relevance scoring and further attempts in applying ontological relations in general domain information retrieval. Meanwhile, some substantial evaluation is planned to compare more baselines and more parameters.

Acknowledgement. This work is supported by National Science Foundation of China (NSFC: 61272233). We thank the anonymous reviewers for the valuable comments.

References

1. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proc. of CIKM 1999, pp. 316–321. ACM, New York (1999)
2. Matsumura, A., Takasu, A.: Adachi: The effect of information retrieval method using dependency relationship between words. In: Proceedings of RIAO 2000, pp. 1043–1058 (2000)
3. Vintar, S., Buitelaar, P., Volk, M.: Semantic relations in concept-based cross-language medical information retrieval. In: Proceedings of ECML/PKDD workshop on Adaptive Text Extraction and Mining (ATEM) (2003)
4. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: Proc. of SIGIR 2004, pp. 170–177. ACM, New York (2004)
5. Morton, T.: Using semantic relations to improve information retrieval. PhD thesis, University of Pennsylvania (2004)
6. Maisonnasse, L., Gaussier, E., Chevallet, J.P.: Revisiting the dependence language model for information retrieval. In: Proc. of SIGIR 2007, pp. 695–696. ACM, New York (2007)
7. Park, J.H., Croft, W.B., Smith, D.A.: A quasi-synchronous dependence model for information retrieval. In: Proc. of CIKM 2011, pp. 17–26. ACM, New York (2011)
8. Hou, Y., Zhao, X., Song, D., Li, W.: Mining pure high-order word associations via information geometry for information retrieval. *ACM Trans. Inf. Syst.* 31(3), 12:1–12:32 (2013)
9. Zhao, J., Huang, J.X., Ye, Z.: Modeling term associations for probabilistic information retrieval. *ACM Trans. Inf. Syst.* 32(2), 7:1–7:47 (2014)
10. Giger, H.P.: Concept based retrieval in classical ir systems. In: Proc. of SIGIR 1988, pp. 275–289. ACM, New York (1988)
11. Lu, X.: Document retrieval: A structural approach. *Inf. Process. Manage.* 26(2), 209–218 (1990)
12. Khoo, C.S.G., Myaeng, S.H., Oddy, R.N.: Using cause-effect relations in text to improve information retrieval precision. *Inf. Process. Manage.* 37(1), 119–145 (2001)
13. Li, Y., Wang, Y., Huang, X.: A relation-based search engine in semantic web. *IEEE Trans. on Knowl. and Data Eng.* 19(2), 273–282 (2007)
14. Lee, J., Min, J.K., Oh, A., Chung, C.W.: Effective ranking and search techniques for web resources considering semantic relationships. *Inf. Process. Manage.* 50(1), 132–155 (2014)
15. Bilotti, M.W., Elsas, J., Carbonell, J., Nyberg, E.: Rank learning for factoid question answering with linguistic and semantic constraints. In: Proc. of CIKM 2010, pp. 459–468. ACM, New York (2010)

16. Voorhees, E.M., Hersh, W.: Overview of the trec 2012 medical records track. In: Proc. of TREC 2012 (2012)
17. Goeuriot, L., Jones, G.J.F., Kelly, L., Leveling, J., Hanbury, A., Müller, H., Salanterä, S., Suominen, H., Zuccon, G.: Share/clef ehealth evaluation lab 2013, task 3: Information retrieval to address patients' questions when reading clinical reports. In: CLEF Online Working Notes (2013)
18. Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., Rindflesch, T.C.: Semmeddb: a pubmed-scale repository of biomedical semantic predications. *Bioinformatics* 28(23), 3158–3160 (2012)

A Unified Microblog User Similarity Model for Online Friend Recommendation

Shi Feng^{1,2}, Le Zhang¹, Daling Wang^{1,2}, and Yifei Zhang^{1,2}

¹ School of Information Science and Engineering, Northeastern University

² Key Laboratory of Medical Image Computing (Northeastern University),
Ministry of Education, Shenyang 110819, P.R.China
{fengshi, wangdaling, zhangyifei}@ise.neu.edu.cn,
zhang7771e@gmail.com

Abstract. Nowadays, people usually like to extend their real-life social relations into the online virtual social networks. With the blooming of Web 2.0 technology, huge number of users aggregate in the microblogging services, such as Twitter and Weibo, to express their opinions, record their personal lives and communicate with each other. How to recommend potential good friends for the target user has been a critical problem for both commercial companies and research communities. The key issue for online friend recommendation is to design an appropriate algorithm for user similarity measurement. In this paper, we propose a novel microblog user similarity model for online friend recommendation by linearly combining multiple similarity measurements of microblogs. Our proposed model can give a more comprehensive understanding of the user relationship in the microblogging space. Extensive experiments on a real-world dataset validate that our proposed model outperforms other baseline algorithms by a large margin.

Keywords: Friend Recommendation, User Similarity, Linear Combination.

1 Introduction

In recently years, people are not satisfied with making friends with their school-mates, colleagues, neighbors and so on. More and more people are willing to extend their real-life social relations into online virtual social networks. Microblogging services, such as Weibo and Twitter, have become very popular, because it allows users to post a short message named tweet or status for sharing viewpoints and acquiring knowledge in real time. According to statistics, by March 2013, there had been 536 million registered users in Sina Weibo and more than 100 million tweets are generated per day in Weibo. Among huge number of users, how to recommend potential good friends for these users has become a critical issue for both commercial companies and research communities.

Different from some traditional social networks, the characteristics of microblog make it more different for finding appropriate friends for the target users. In microblog, the users can follow someone without his or her permissions. Therefore,

the friend links are more casual and informal in microblog than in other online social networks. The users may add a friend link to someone because they share similar hobbies, have similar tags, live nearby, have been to the same places or they have similar opinions and have just discussed about the same trending topics in microblog. These characteristics have posed severe challenges for potential good friend recommendation in microblog.

In this paper, we propose a novel unified microblog user similarity measurement model for online friend recommendation. Our proposed model can integrate multiple similarity measurements of microblog together by linear combination and learn corresponding weight for each measurement. As a result, we can provide a more comprehensive understanding of users' relationship in the microblogging space and recommend potential good friend for the target user. To summarize, the main contributions of our work are as follows.

- (1) We leverage the massive real-world microblog data to analyze the characteristics of microblog features and determine which features are critical for friend recommendation.

- (2) We proposed a microblog user similarity model for friend recommendation by linearly combining multiple similarity measurements of microblogs.

- (3) We conduct extensive experiment on a real-world dataset. The experiment results validate the effectiveness of our proposed model and algorithm.

The structure of the rest of the paper is as follows. Related work is discussed briefly in Section 2. In Section 3, we analyze the crawled dataset and introduce the characteristics of the friend relationship in microblog. In Section 4, we propose the unified microblog user similarity measurement model. Section 5 introduces the settings and details of the experiments. We compare our proposed model with the other baseline models. In Section 6, we make a conclusion and point out the directions for our future work.

2 Related Work

Potential friend recommendation in social network is a hot research topic in the academic area. Guo et al. utilized the tag trees and relationship graph to generate the social network and employed the network topology and user profile to recommend friends [1]. Chin et al. focused on the friend recommendation in Facebook and LinkedIn. They considered the user daily behaviors as good features to indicate potential good friends for the target users [2]. Shen et al. leveraged three dimensions Utilitarian, Hedonic and Social to explore the recommendation model in the mobile terminals [3]. Yu et al. built heterogeneous information networks and transition probability matrix [4]. They conducted random walk on the built graph to recommend friends for the target users. Sharma et al. studied on the value of ego network for friend recommendation [5]. Chu et al. studied on the effect of location information in mobile social network for friend recommendation [6]. The tag-based and content-based friend recommendation algorithms were compared in [7]. The authors described a comprehensive evaluation to highlight the different benefits of tag-based and content-based recommendation strategies.

Silva et al. analyzed the structure of the social network and utilized the topology of the sub-graphs to recommend potential friends [8]. Their algorithm significantly outperformed the traditional Friend-of-Friend method that is also a topology based algorithm. In [9], Moricz et al. introduced the friend recommendation algorithm People You May Know. Not only the precision, but also the speed of the algorithm was considered for the algorithm in MySpace. Bian et al. presented a collaborative filtering friend recommendation system MatchMaker based on personality matching [10]. The authors collected feedback from users to do personality matching, which provided the users with more contextual information about recommended friends.

Although a lot of papers have been published for friend recommendation, most of the existing literature focused on unique feature for recommendation. Actually, the potential good friends for a target user can be affected by multiple features, which is the basic assumption of this paper.

3 The Characteristics of Friend Relationship in Microblogs

Who is the target user's potential good friend in microblogs? It can be determined by many features because the microblog is full of personal and social relation information. To analyze the characteristics of friend relationship in microblogs, we have crawled huge number of microblog users from Weibo, which is the largest microblogging service in China. The statistics of our crawled dataset is shown below.

Table 1. The statistics information of the crawled dataset

Dataset Features	NO. of Features	Percentage of the Whole Dataset
Independent Users	1,459,303	--
Friend Links	3,853,864	--
Bi-directional Friend Links	9,646	--
Users with Tags	1,017,443	69.7%
Users with Location Information	1,292,942	88.6%
Users with Check-in Information	457,520	31.4%
Users with Hot topics	537,741	36.8%

In this paper, we define a friend link from user A to user B if user A follows user B in the microblog, and we say user B is a friend of user A. Due to the characteristics of Weibo, a friend link from user A to B does not necessarily mean that there is a link from user B to A. We have crawled more than 1.4 million microblog users with 3.8 million friend links and there are only about 9,646 links are bi-directional.

Besides the friend relationship, there are a lot of social information in microblogs, such as tags, location and check-in information. The tags are a set of key words that can describe the professions, interests and hobbies of the user. The location information describes the city where the user lives in. The check-in information denotes the GPS location that the user has been to. The hot topics are the trend topics

in microblogs that the user involved in. We compare the similarity of the users using the above features respectively and the results are shown in Table 2.

Table 2. The average similarity between users of friends and strangers

	Tag	Location	Check-in	Hot topic
Friends	4.4×10^{-3}	0.082	0.037	3.0×10^{-4}
Strangers	1.6×10^{-3}	0.038	0.022	2.3×10^{-4}

We can see from Table 2 that in the crawled dataset, the average similarity between friends calculated by specified feature is much higher than the average similarity between strangers. Based on these observations, we assume that tag, location, check-in and hot topic features are good indicators for friend recommendation. In the next section, we will recommend potential good friend for microblog users by linear combination of the multiple similarity measurements.

4 Friend Recommendation by Combining Multiple Measures

In the above section, we observe that tag, location, check-in and hot topic information are good indicators for users’ interests, hobbies and professions in microblogs. In this section, we propose a linear combination approach to integrate these different similarity measurements together for friend recommendation. The overall framework of our proposed approach is shown in Figure 1.

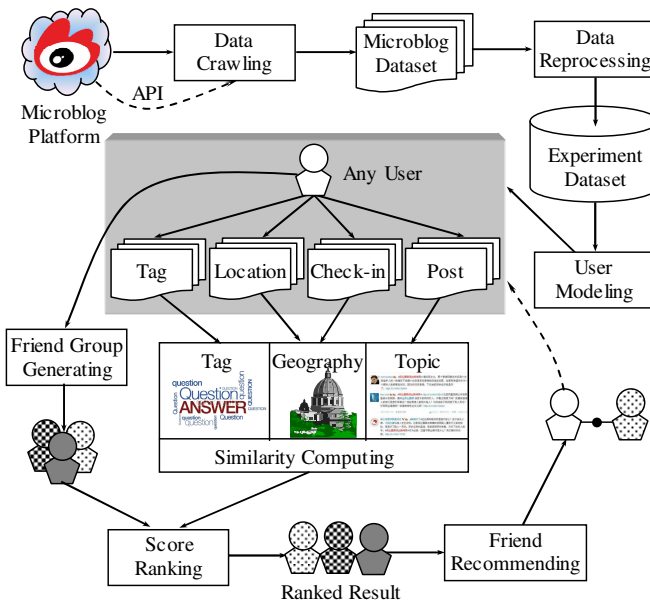


Fig. 1. The overall framework of the proposed unified friend recommendation model

4.1 Candidate Friend Set Generation

Due to the huge number of users in the microblogs, we can not traverse the whole microblogging space to find whether a user is a potential good friend for the target user. The detail of our candidate friend set generation algorithm is shown below. Generally speaking, in line 3-6 we select the friends of the user's friends into the candidate set. In line 7, we add the most popular users that are usually celebrities into the candidate friend set.

Algorithm 1. Generation of user's candidate friend set;

Input: The current friend list of users, the number of followers of users, the number of friends who will be recommended k ;

Output: The candidate friends set of target user u ;

Description:

1. FOR every target user u
 2. Extract u 's current friends set $f(u)$, $f(u) = \{u_1, u_2, \dots, u_n\}$.
 3. FOR every user u_i IN $f(u)$:
 4. Extract u_i 's current friends set $f(u_i)$.
 5. Generate u 's first candidate friends set $f_r(u)$ from the current friends set of u and his friends set, $f_r(u) = \bigcup_{i=1}^n f(u_i) - f(u)$.
 6. Select top- k users from $f_r(u)$ to build the second candidate friends set $f_{r'}(u)$ according to the number of common friends between the target user u and the candidate friend u_i .
 7. Add k most popular users who have most followers and are not the current friends of u into the second candidate friends set to build the final candidate friends set $f_c(u)$ which contains $2k$ users.
-

4.2 User Tag Similarity

Tags are words or phrases that users utilize to describe online resources. These tags can indicate users' personal interests and hobbies. However, it is usually difficult to directly calculate the similarity between tags because many tags are out of the knowledge base such as WordNet and user-tag vectors are very sparse. Directly using cosine function and tags as vectors may lose many potential information for user similarity calculating. To tackle these challenges, in this paper we employ hierarchical clustering algorithm [11] to build the tags as a tree, based on which the user tag set similarity is calculated. The main steps of our proposed method are discussed as follows.

- (1) Given the tag set T that extracted from all the users of the crawled dataset, eliminate the tags that have very low occurrence frequencies in T .
- (2) Partition the tags in T based on hierarchical clustering algorithm and build a tag set tree, such as the example in Figure 2. The similarity between tags is calculated by their co-occurrences in T .
- (3) Recalculate the personalized tag set similarity based on the tag tree.

If two tags do not co-occur in the T , or they are out of knowledge base such as WordNet, we could not calculate their similarity directly. In Figure 2, the root node contains all the tags in T , and the leaf node contains only one tag. It is obvious that the similarity between the two tags is bigger if there are fewer hops between them in tag tree. When there are the same hops, if the depth of the tags gets bigger, they become more similar with each other. The depth means the number of hops to the root node.

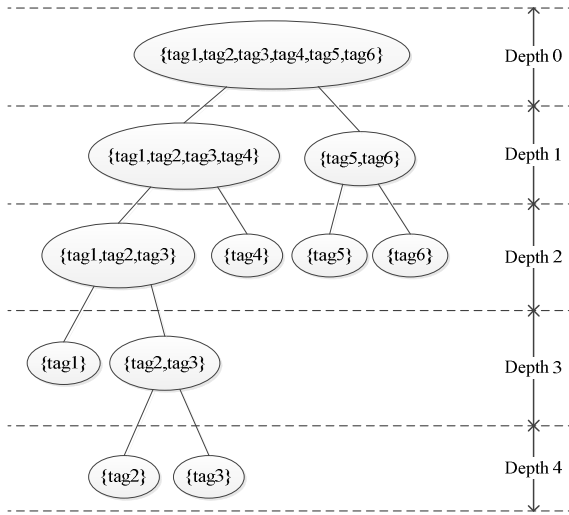


Fig. 2. The example of a tag tree

Given two user u_i, u_j , and their corresponding tag set $T(u_i)=\{t_1, t_2, t_3, \dots, t_m\}$, $T(u_j)=\{t_1, t_2, t_3, \dots, t_m\}$, $t_a \in T(u_i)$, $t_b \in T(u_j)$. The similarity between t_a and t_b can be calculated by the tag tree as:

$$sim_t(t_a, t_b) = \frac{1}{\sum_{k, k+1 \in SP(a,b)} 2^{l(d(k) + d(k+1))}} \tag{1}$$

where a, b represent the node of t_a and t_b in tag tree respectively; $SP(a,b)$ denotes the nodes that in the shortest path between a and b ; $d(k)$ denotes the depth of the node k in the tag tree. Therefore, the personalized tag similarity between u_i and u_j is calculated by the average similarity between $T(u_i)$ and $T(u_j)$ as:

$$sim_{ts}(u_i, u_j) = \frac{1}{|T(u_i)||T(u_j)|} \sum_{t_a \in T(u_i)} \sum_{t_b \in T(u_j)} sim_t(t_a, t_b) \tag{2}$$

We normalized the value of sim_{ts} between users in the candidate friend set for the further calculation.

4.3 User Geography Similarity

Usually, the user's online friendship is a virtual reflection of the real world relationship. We observe that if the users have the same living city and usually come to the similar places, they intend to be friends with each other. In this section, we calculate the geography similarity of microblog users based on their location and check-in information.

Location Similarity. We utilize $sim_{ct}(u_i, u_j)$ to represent the location similarity between user u_i and u_j . If two users have the same location value, $sim_{ct}(u_i, u_j)=1$; Otherwise, $sim_{ct}(u_i, u_j)=0$.

Check-in Similarity. The Weibo platform has divided users' check-in information into twelve categories, such as "Train Station", "Library", "School" and so on. Therefore, each user's check-in information is represented by a vector with 12 dimensions, i.e. $chk(u)=\{cp_1, cp_2, \dots, cp_{12}\}$, where cp_i is the proportion of the number of check-in category i in the latest 50 check-ins. We can use cosine function to calculate their similarity. So the check-in similarity between two users is calculated by $sim_{chk}(u_i, u_j)=\cos(chk(u_i), chk(u_j))$.

Finally, the geography similarity between users in microblogs is calculated by:

$$sim_{loc}(u_i, u_j) = \gamma \cdot sim_{ct}(u_i, u_j) + (1 - \gamma) \cdot sim_{chk}(u_i, u_j) \quad (3)$$

where γ is the weight parameter. We will finally normalize the geography similarity for further similarity linear combination.

4.4 User Hot Topic Similarity

Usually users like to talk about the hot topics in microblog. The hot topic discussion that user takes part in could reflect his/her interests and hobbies. For user u_i , we employ Weibo API to extract the hot topics that u_i takes part in, and the extracted topic set is represented by $TP(u_i)$. The hot topic similarity between u_i and u_j is calculated by Jaccard similarity as:

$$sim_{tp}(u_i, u_j) = Jaccard(TP(u_i), TP(u_j)) = \frac{|TP(u_i) \cap TP(u_j)|}{|TP(u_i) \cup TP(u_j)|} \quad (4)$$

In Formula 4, if two users have discussed more hot topics in common, they will have bigger similarity. We will finally normalize the hot topic similarity for further similarity linear combination.

4.5 A Unified Microblog User Similarity Model

In Section 3, we observe that the tag, location, check-in, and hot topic information are all good indicators for friend recommendation in microblog. In this paper, we propose a unified microblog user similarity model by linearly combining the multiple similarity measurements. Given a user u , and a user u_i from the candidate friend set of u , i.e. $u_i \in f_c(u)$, we have the unified similarity function:

$$sim(u, u_i) = \alpha \cdot sim_{ts}(u, u_i) + \beta \cdot sim_{loc}(u, u_i) + (1 - \alpha - \beta) \cdot sim_{tp}(u, u_i) \quad (5)$$

where u_i is a candidate friend of u generated by Algorithm 1; sim_{ts} , sim_{loc} , and sim_{tp} are tag, geography and hot topic similarity respectively; α and β are weight parameters for the linear combination.

Given a target user u , we first employ Algorithm 1 to generate the candidate friend set $f_c(u)$. Then we traverse $f_c(u)$ to calculate the similarity between the candidate friend and the target user u . The top ranked K users will be extracted as recommended friends for the target user u .

5 Experiment

5.1 Experiment Setup

Our experimental dataset are crawled from Sina Weibo platform using API tool [12]. The detail statistics of the crawled dataset is show in Table 1 of Section 3. We conduct experiments using a PC with Inter Core i7, 8 GB memory and Windows 7 as the operation system.

We employ 5-fold cross validation to conduct the experiments. For each target user u , we randomly partition u 's current friends and non-friends into 5 groups respectively. We randomly put one group of friends and one group of non-friends together to form a subset of the crawled data. For each run, four of the five subsets are used for training the parameters in Formula 5 and the remaining one subset is used for testing. We utilize Precision, Recall and F-measure to evaluate the performance of the proposed model and algorithms.

5.2 Experiment Results

Firstly, we learn the parameter γ for the Formula 3. In this experiment, we only utilize the geography similarity to recommend friends for the target user. The experiment results are shown from Figure 3 to Figure 5.

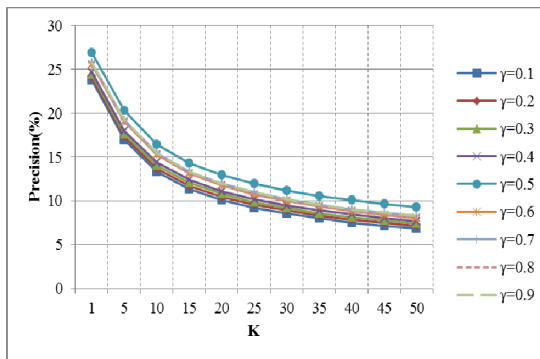


Fig. 3. The result of the parameter selection of the geography similarity model (Precision)

We can see from Figure 5 that when the number of recommended friends K gets bigger, the F-Measure of the model firstly increases dramatically and then gradually decreases. The best performance is achieved when $K=10$ and for all K settings, we can get the best performance using $\gamma=0.5$. Therefore, for the following experiments, we set $\gamma=0.5$.

Secondly, we learn the parameter α and β for the Formula 5. In Formula 5, we unified tag, location, check-in and hot-topic information together by linear combining. The experiment result is shown in Figure 6.

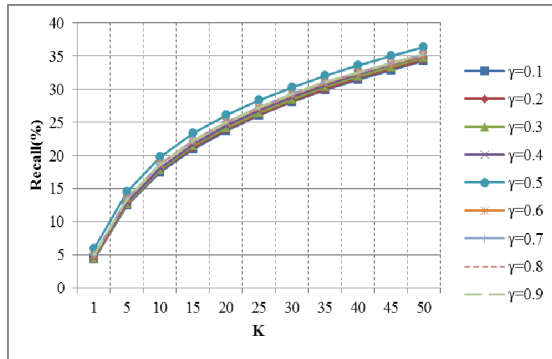


Fig. 4. The result of the parameter selection of the geography similarity model (Recall)

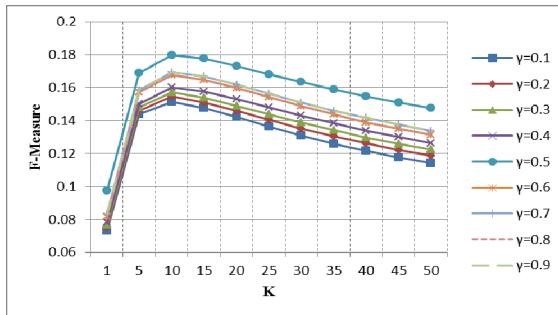


Fig. 5. The result of the parameter selection of the geography similarity model (F-measure)

In Figure 6, when α and β are small, the F-Measure of the unified model is relatively small. When α is fixed, F-Measure grows bigger as β grows. When β is fixed, F-Measure grows bigger as α grows. The best performance is achieved when $\alpha=0.4$ and $\beta=0.5$, which are used as default settings for the following experiments.

To evaluate the effectiveness of the proposed unified model, we compare our method with some other friend recommendation algorithm.

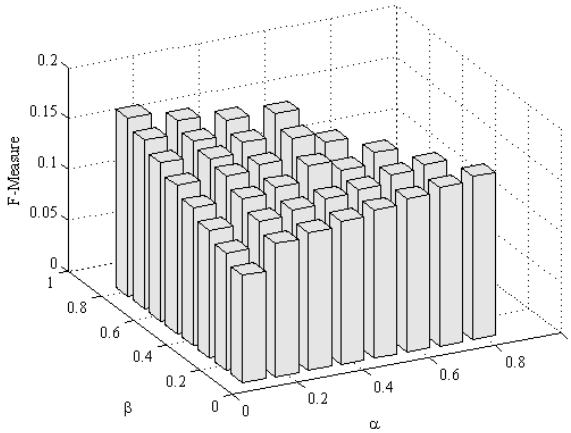


Fig. 6. The result of the parameter selection of the unified friend model

(1) FOF+ (Friends of Friends+). This method utilizes Algorithm 1 in Section 4.1 to generate the friends of the target user.

(2) FC (Follower Count). FC algorithm uses the number of followers to rank the candidate friends. The top K ranked users are extracted as the recommended friends.

(3) FOF+TS (Friends of Friends + Tag Similarity). This method utilizes Algorithm 1 to generate candidate friends set and employ the tag similarity to extract top K ranked users as recommended friends.

(4) FOF+LS (Friends Of Friends + Geography Similarity). This method utilizes Algorithm 1 to generate candidate friends set and employ the geography similarity to extract top K ranked users as recommended friends.

We denote the proposed Unified Microblog Friend Recommendation model as UMFR. We compare UMFR with above baseline methods, and the details are shown in Figure 7, Figure 8, and Figure 9.

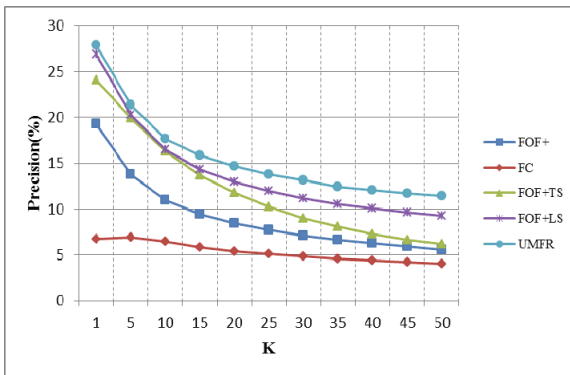


Fig. 7. The result of the friend recommendation algorithms (Precision)

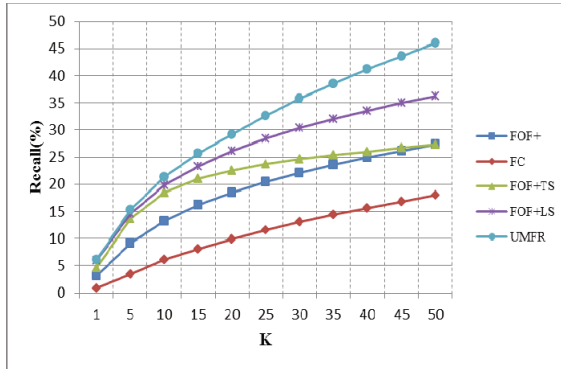


Fig. 8. The result of the friend recommendation algorithms (Recall)

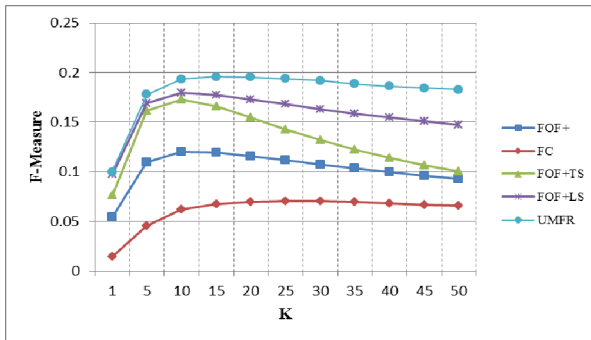


Fig. 9. The result of the friend recommendation algorithms (F-measure)

In Figure 7, the X-axis represents different number of the recommended friends; the Y-axis represents the precision of the algorithms. From Figure 7, we can see that when there are more recommended friends, the precisions of all the algorithms decrease gradually. With the same K setting, when K is small, there is no significant difference between our proposed method and geography based methods. The proposed method significantly outperforms the FOF+ and FC methods. This is because when K is small, the user with similar tags and locations are easily selected from the candidate friend set. The social relations and number of followers do not play a critical role in this case. When K gets bigger, our proposed method’s precision has a smoother decline curve. This is because our proposed UMFR model that considering multiple feature measures provides a more comprehensive measurement for user similarities. The precision of FC method does not change much according to the K values, but obviously it has the worse performance of all the algorithms.

In Figure 8, the Y-axis represents the Recall of the algorithms. We can see from Figure 8 that the Recall of UMFR, FOF+LS and FOF+TS are similar to each other. However, when K gets bigger, our proposed method has better and better performance than the other compared algorithms.

In Figure 9, the Y-axis represents the F-Measure of the algorithms. We can see from Figure 9 that as K grows, the F-Measure of all the algorithms firstly get dramatically increases, and then drop down gradually. Our proposed UMFR model significantly outperforms other baseline methods and achieves the best performance when K is between 10 and 15.

6 Conclusions and Future Work

Recently, people are willing to make friends in the online social networks. Because of the multiple features, the traditional friend recommendation algorithms fail to capture the characteristics of the microblogs for user similarity measurement. In this paper, we propose a novel unified microblog user similarity measurement model for online friend recommendation. Our proposed model linearly combines multiple similarity measures of the users, which provides a comprehensive understanding of the user relationship in microblogs. Experiment results show that our proposed method significant outperforms the other baseline methods. Future work includes integrating more microblog features for measuring similarity between users for improving the quality of friend recommendation. We also intend to take the time factor into account, so that we can recommend different potential friends at different time.

Acknowledgements. This work is supported by the State Key Development Program for Basic Research of China (Grant No. 2011CB302200-G), State Key Program of National Natural Science of China (Grant No. 61033007), National Natural Science Foundation of China (Grant No. 61100026, 61370074, 61402091), and Fundamental Research Funds for the Central Universities (N120404007).

References

1. Gou, L., You, F., Guo, J.: SFViz: Interest-based friends exploration and recommendation in social networks. In: Proceedings of the Visual Information Communication-International Symposium (2011)
2. Chin, A., Xu, B., Wang, H.: Who should I add as a friend?: A study of friend recommendations using proximity and homophily. In: Proceedings of the 4th International Workshop on Modeling Social Media (2013)
3. Shen, X., Sun, Y., Wang, N.: Recommendations from friends anytime and anywhere: Toward a model of contextual offer and consumption values. *Cyberpsychology, Behavior, and Social Networking* 16(5), 349–356 (2013)
4. Yu, X., Pan, A., Tang, L.: Geo-friends recommendation in gps-based cyber-physical social network. In: Proceedings of International Conference on Advances in Social Networks Analysis and Mining, pp. 361–368 (2011)
5. Sharma, A., Gemici, M., Cosley, D.: Friends, strangers, and the value of ego networks for recommendation. In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, pp. 721–724 (2013)
6. Chu, C., Wu, W., Wang, C.: Friend recommendation for location-based mobile social networks. In: Proceedings of Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 365–370 (2013)

7. Hannon, J., McCarthy, K., Smyth, B.: Content vs. Tags for Friend Recommendation. In: Research and Development in Intelligent Systems XXIX, pp. 289–302. Springer, London (2012)
8. Silva, N., Tsang, I., Cavalcanti, G., Tsang, I.: A graph-Based friend recommendation system using genetic algorithm. In: Proceedings of IEEE World Congress on Computational Intelligence, pp. 233–239 (2010)
9. Moricz, M., Dosbayev, Y., Berlyant, M.: PYMK: Friend Recommendation at MySpace. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 999–1002 (2010)
10. Bian, L., Holtzman, H.: Online friend recommendation through personality matching and collaborative filtering. In: Proceedings of the Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 230–235 (2011)
11. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
12. Weibo API, <http://api.weibo.com>

Weakly-Supervised Occupation Detection for Micro-blogging Users

Ying Chen¹ and Bei Pei²

¹ China Agricultural University, China, 100083
chenying@cau.edu.cn

² Key Lab of Information Network Security, Ministry of Public Security, China, 200031
peibei@stars.org.cn

Abstract. In this paper, we propose a weakly-supervised occupation detection approach which can automatically detect occupation information for micro-blogging users. The weakly-supervised approach makes use of two types of user information (tweets and personal descriptions) through a rule-based user occupation detection and a MCS-based (MCS: a multiple classifier system) user occupation detection. First, the rule-based occupation detection uses the personal descriptions of some users to create pseudo-training data. Second, based on the pseudo-training data, the MCS-based occupation detection uses tweets to do further occupation detection. However, the pseudo-training data is severely skewed and noisy, which brings a big challenge to the MCS-based occupation detection. Therefore, we propose a class-based random sampling method and a cascaded ensemble learning method to overcome these data problems. The experiments show that the weakly-supervised occupation detection achieves a good performance. In addition, although our study is made on Chinese, the approach indeed is language-independent.

Keywords: occupation detection, sampling and ensemble learning.

1 Introduction

Micro-blogging platforms such as Twitter and Plurk, not only provide services for users to share information with friends, but also contain plenty of personalization business applications which usually are designed according to users' personal information (i.e. education experience, working experience and hobby). However, as personal information is usually not obligatorily provided by users, this kind of information is often incomplete or omitted. In this paper, we attempt to explore how to automatically detect personal information for micro-blogging users. To better explain our work, in this paper, we adopt the twitter's terminology. A "tweet" refers to a short message a user shares with others; a "following" is a user who the focused user is subscribed to; a "follower" is a user who subscribes to the focused user.

As far as we know, there have been few studies on personal information detection for micro-blogging users. Although intensive studies on personal information detection have been done in the past years, most of them focus on factual (objective) texts,

such as news reports and homepages. Since those factual texts tend to introduce the focused person, the personal information is often given in the context of the mentions of the person of interest. In contrast, tweets are more like a type of subjective texts. Instead of introducing themselves, micro-blogging users like to express their opinions or describe their activities in their tweets. Therefore, those tweets sometimes do not provide enough explicit personal information, and personal information detection requires more deduction.

As personal information is rather broad, in this paper, we focus only on occupation information. In fact, occupation information can be considered as a hierarchical structure. The first level roughly contains three types of occupations (student, employee and un-employed), and each occupation then is further divided in the next level. In this paper, we explore occupation detection only according to the first-level division, and examine the research issues specifically for micro-blogging user.

In this paper, we first explore how to effectively integrate the micro-blogging information of a user (i.e. tweets and personal descriptions) for occupation detection. Furthermore, we investigate how to deduce the occupation of a user with the aid of some particular tweets of the user. Overall, there are three contributions of our user occupation detection.

First, we automatically construct a user occupation corpus with a set of rules and the rules can infer the occupations of some users according to their personal descriptions if exist. Although this user occupation corpus is very noisy, it can avoid costly human annotation and give a guide to the design of our user occupations detection. From the user occupation corpus, we find that few users explicitly release their un-employed status, and thus, we formulate the user occupation detection as a classification problem with three classes (student, employee and undetermined). Here, “undetermined” users refer to the ones whose occupations cannot be inferred from the given micro-blogging information even by humans.

Second, given the pseudo-annotated user occupation corpus, the following approach is intuitive for our user occupations detection: a supervised classification method (such as SVM, the Maximum Entropy model and so on) is chosen, and the features of an instance are extracted from all tweets of a user. However, this intuitive approach cannot work well because of the three data problems inherited in our pseudo-annotation corpus: data imbalance, data bias and data noise. The data imbalance refers to the imbalance between the three classes, and requires a specific classification approach (i.e., imbalance classification). The data bias refers to the class distribution in our corpus does not follow the real one. The data noise refers to noisy features and noisy pseudo tags in our corpus. In this paper, to overcome the data imbalance and the data bias, we choose a typical imbalance classification approach, which uses MCS (a multiple classifier system [12]) and a sampling method. The sampling method selects the balanced training datasets for the base classifiers of MCS. Furthermore, because the data noise is severe, we find that the typical sampling methods, such as random over-sampling [5] and random under-sampling [4], cannot perform well for our task. Thus, we propose a class-based random sampling method, which is an extension of random under-sampling. The empirical experiments show that our MCS-based user occupation detection system achieves a good performance.

Third, users are much different in terms of the scales of their tweets. For example, some users post several tweets and some have thousands. In fact, we observe that the occupation of a user can be determined only by several occupation-related tweets. Thus, it is better to detect user occupation only according to this kind of tweets. Unfortunately, how to select the occupation-related tweets is also a hard task. In this paper, we propose a cascaded ensemble learning method which selects some occupation-related tweets and uses them to further improve the user occupation detection.

2 Related Work

In this section, we first compare our user occupation detection with previous works on personal information detection, and then briefly present the state-of-the-art studies on imbalanced classification.

2.1 Occupation Detection

For occupation detection, several systems are presented in the bakeoff, searching information about entities in the Web (WePS). WePS works on the occupation detection for a web person, which extract the occupation information of a person from the given webpages. Artiles et al. [2] summarize these systems and find that a rule-based approach [6] is most effective because the approach can capture the structures of some kinds of webpages.

Furthermore, occupation detection can be considered as a sub-problem of Information Extraction (IE). The survey of Sarawagi [17] examines rule-based and statistical methods for IE, and point out that the different kinds of approaches attempt to capture the diversity of clues in texts [1,3,7,16]. Therefore, the properties of texts determine the approaches of occupation detection.

For the occupation detection for micro-blogging users, there are two types of textual information: personal descriptions and tweets. Moreover, these two types of textual information are complement for the user occupation detection. A personal description is a kind of structured texts, and occupation detection on those structured texts is well studied. In fact, the main challenge comes from tweets because tweets have their own characteristics, such as informal expressions, short texts and so on. In this paper, we explore the interaction of these two types of textual information for occupation detection.

2.2 Imbalanced Classification

The common understanding for data imbalance for multi-class classification is that the imbalance exists between the various classes. Because of severe class distribution skews, in most cases, classifiers trained with imbalanced data prefer to annotate test instances with majority class (MA) and ignore the minority class (MI). Thus, imbalanced data requires specific approaches, namely imbalanced classification.

Imbalanced classification has been widely studied in terms of data level and algorithmic level (see the comprehensive review [10]). From the data level, the most important approach is sampling which attempts to balance the class distribution, such as various over-sampling methods that replicate MI instances [5,8-9,18] and various under-sampling methods that remove MA instances[4,11,15,19]. From the algorithmic level, many approaches are proposed, such as cost-sensitive learning, one-class learning, and ensemble learning.

In this paper, we attempt to use a sampling method to solve both the data imbalance and the data bias in our corpus. Although the empirical study [13] shows that under-sampling is most effective for sentiment classification, it does not work well for our task because of the severe noise in our corpus. Thus, we explore how to extent the under-sampling method so that to handle the noisy data.

3 The User Occupation Corpus

In this section, we first introduce the construction of our user occupation corpus (including the corpus collection and the rule-based user occupation detection), and present an in-depth corpus analysis. According to the analysis, we then formulate our user occupation detection task.

3.1 The Corpus Collection

Our user occupation corpus indeed contains a set of users and a user has an information unit containing all information regarding the user (tweets, followings, followers, and personal descriptions). The whole corpus is collected through the following four steps.

1. Two hot topics are chosen from the Sina micro-blogging platform (a Chinese micro-blogging website): one is “College English Test” (a student activity), and the other is “the symptoms for going to work on Monday” (an employee activity). A user who posts a tweet for either of the two topics is considered as a seed. There are totally ~1,800 seeds.
2. Initial a user set which contains all seeds.
3. Beginning with the initial user set, the user set is iteratively increased by incorporating their friends. Here, a “friend” of a focused user refers to a user who is both a following and a follower of the user.
4. For each user in the user set, all related information is crawled from the Sina micro-blogging platform.

Although our corpus is scalable through the iteration of Step 3, due to the limited time, our corpus collect only totally 30,840 users, which is ~0.015% of the whole Sina micro-blog users. However, we can still gain enlightenments on the user occupation detection through our pilot study on this rather small-scale corpus.

3.2 The Rule-Based User Occupation Detection

Because human annotation is time-costing, in this paper, we propose a rule-based user occupation detection system, which automatically detects the occupations of some users according to their personal descriptions if exist.

Although the Sina micro-blogging platform provides templates for users to input their occupation information, we observe that users often do not exactly follow them. For example, a user lists only his working company “Lenovo” without time information. Because of the incompleteness, the rule-based user occupation detection becomes challenging.

In the rule-based user occupation detection, for a user, his/her working experience firstly is examined. If the job information is provided, the user is tagged as “employee”. Otherwise, go to next. Secondly, the education experience is examined. If the college information is provided, the user is tagged as “student”. Otherwise, the user is tagged as “undetermined”.

3.3 The Corpus Analysis

In our user occupation corpus, ~74% instances (users) provide their personal descriptions, and however, only ~36% instances prefer to publish their occupation information. Furthermore, our rule-based occupation detection detects the occupations only for 31% users (~17% are students and ~14% are employees). This indicates that only some of Sina micro-blogging users present useful occupation information through their personal descriptions.

After the rule-based occupation detection, a user in our corpus is either an instance with an occupation (namely, a rule-determined instance, which is either a student or an employee) or an “undetermined” instance (namely, a rule-undetermined instance, whose occupation cannot be detected by our rules). In the following section, we examine the rule-determined data and the rule-undetermined data, which contains all rule-determined instances and all rule-undetermined instances, respectively.

The rule-determined data: for the rule-determined data, we reserve ~1000 instances for the development data (namely, the rule-determined dev) and ~1000 instances for the test data (namely, the rule-determined test). These two datasets then are annotated by humans as follows.

An instance is tagged with one of the four tags: student, employee, un-employed and undetermined. For a user, an annotator reads his/her tweets one by one in chronological order (beginning with the most recent tweet). If a tag can be confidently given to the user according to the present tweet, the annotator stops. Finally, if the annotator cannot assign a tag after reading all tweets of the focused user, the “undetermined” tag is given to the user.

According to the human-annotated data, we find that the overall accuracy of the two rule-determined datasets is ~72%. This indicates that ~72% users deliver the same occupation information both in their personal descriptions and in their tweets. Moreover, the real occupation distribution in the two rule-determined datasets is: student (50.8%), employee (36.5%), un-employed (1.2%) and undetermined (11.5%).

The rule-undetermined data: for the rule-undetermined data, we reserve ~1000 instances for the development data (namely, the rule-undetermined dev) and ~1000 instances for the test data (namely, the rule-undetermined test). Similar to the rule-determined data, these two datasets are annotated by humans. For the two rule-undetermined datasets, the overall accuracy is ~28%, and the real occupation distribution is: student (40.2%), employee (31.4%), un-employed (0.4%) and undetermined (28.0%).

3.4 Task Formulation

According to the real occupation distribution for the rule-determined data and the rule-undetermined data, we formulate the user occupation detection task as follows. Because the tag “un-employed” occupies such a low percentage (~1%) that we decide to ignore them in our current work, the occupation detection becomes a classification problem with three classes (student, employee, and undetermined). The low percentage of “un-employed” may be due to the fact that most of un-employed users do not like to mention their job status.

Regarding the data setting for our user occupation detection, except for the human-annotated instances in Section 3.3, we use all instances with the tags outputted from the rule-based user occupation detection as the training data (namely, the pseudo-training data). In the pseudo-training data, 15.2% instances are students, 12.6% are employees, and 69.2% are undetermined instances.

4 The MCS-Based User Occupation Detection

In this section, we first examine the data problems in the pseudo-training data, and then introduce the MCS-based user occupation detection. Particularly, we present our class-based random sampling method and cascaded ensemble learning method.

4.1 The Overview of the MCS-Based User Occupation Detection

Our user occupation detection is a three-class classification problem. Given the pseudo-training data, many common supervised classification methods cannot work effectively because of the following three data problems.

1. Data imbalance: the data imbalance problem is severe in the pseudo-training data. For example, the imbalance ratio between “undetermined” and “employee” is ~4.6 (69.2% vs. 15.2%) although only some “undetermined” instances are real undetermined by humans.
2. Data bias: our user occupation corpus is somewhat biased to the users with occupations (“student” or “employee”) because of the selection of topics during the corpus collection (see Section 3.1). Thus, the pseudo-training data may not reflect the real occupation distribution even it can be annotated by humans. In the other hand, it is difficult to capture the real occupation distribution of micro-blogging users because it is changeable.

3. Data noise: there are two kinds of data noises in the pseudo-training data: noisy features and noisy pseudo tags. First, micro-blog is popular because it has fewer restrictions on writing styles. Thus, tweets themselves are intrinsically noisy, and furthermore, the features based on tweets are likely to be noisy. Second, since the rule-based user occupation detection achieves only a decent performance ($\sim 72\%$ for the rule-determined data and $\sim 28\%$ for the rule-undetermined data, see Section 3.3), the tags in the pseudo-training data are severely noisy, particularly for “undetermined”.

Although data imbalance and data bias seem different from each other, both of them involve the skewed class distribution in the pseudo-training data, and they can be solved with the same approach – a sampling method which can select a balanced training dataset for a classifier. After taking the data noise into account, we propose a class-based random sampling method. Moreover, considering that users have various-scale tweets, we propose a cascaded ensemble learning method which integrates the occupation information of all tweets and the occupation information of individual tweets to do user occupation detection.

In this paper, we choose MCS as the framework of our system. There are two stages in MCS: training and test. During the training stage, a base classifier is trained with a supervised classification method as well as some training instances (users) selected by the class-based random sampling method. For each training instance, all of the tweets are catenated into a document on which feature extraction works. During the test stage, the cascaded ensemble learning method is used. The class-based random sampling method and the cascaded ensemble learning method are described as follows.

4.2 The Class-Based Random Sampling

Random under-sampling is a typical sampling method for imbalance classification. It randomly selects a subset of the MA instances from the initial training data and then combines them with all of the MI instances to form the training dataset for a base classifier. Our analysis shows that random under-sampling or its variations perform well because they satisfy the following two conditions: (1) the MI instances are high-quality, such as human-annotated data [13]; (2) all of the MI instances are used in a base classifier. Unfortunately, our pseudo-training data indeed is very noisy, and a base classifier will be confused if its training dataset includes all of the MI instances. In this paper, we propose a class-based random sampling method.

In general, the class-based random sampling method (shown in Figure 1) guarantees that the instances belonging to the same class are equally likely to be chosen. The class-based random sampling has only one parameter K , and our empirical experiments (see Section 5.2) show that the parameter has a big impact to the effect of the sampling. In addition, random under-sampling is actually an extreme case of the class-based random sampling method where K is chosen the maximum.

Although our class-based random sampling looks simple, it can effectively solve the aforementioned three data problems. First, the training dataset for a base classifier

is balanced because each class contributes exact K instances. Thus, the problem of data imbalance and data bias can be avoided. Second, our class-based random sampling selects different subsets of the initial training data for base classifiers. Given the noisy pseudo-training data, the more possible the subsets are, the more possible it is for an effective feature to be selected by a base classifier. This is also the fundamental difference between our class-based random sampling and various under-sampling methods.

4.3 The Cascaded Ensemble Learning

We propose a cascaded ensemble learning method, as shown in Figure 2. In general, there are two steps. Firstly, for a test user, two tags are gotten from the two ensemble learning methods, the whole-tweets-based ensemble learning and the individual-tweet-based ensemble learning. Secondly, according to the two tags, a rule-based ensemble learning is used to get the final tag of the test user. The three ensemble learning methods are explained as follows. Notice, “user” and “instance” are not exchangeable in this section.

The whole-tweets-based ensemble learning: it is very simple ensemble learning based on the majority vote. Firstly, for a test user, a set of tags are obtained from the base classifiers. Secondly, the majority vote is applied to the set of tags to get the final tag of the test user. Notice, similar to the training of the base classifiers, a test instance inputted to a base classifier is a document which contains all tweets posted by the test user.

The individual-tweet-based ensemble learning: although different users post various-scale tweets, it is often a case that the occupation of a user is determined only by several occupation-related tweets. Therefore, we propose an individual-tweet-based ensemble learning, explained as follows. Step 1 and 2 attempt to select some occupation-related tweets, and Step 3 and 4 try to use the occupation-related tweets for user occupation detection.

1. Given a test user, detect the occupation tag for each of his/her tweets. The procedure is similar to the whole-tweets-based ensemble learning except the following two things: a test instance inputted to a base classifier is an individual tweet, and the tag is attached with the support rate which is calculated during the majority vote.
2. For each individual tweet, if its tag is “student” or “employee”, examine its support rate. If the support rate is greater than the given threshold, the individual tweet is considered as an occupation-related tweet. Since the base classifiers do not perform very well, we treat only the tweets whose tags have high confidences as occupation-related tweets.
3. If the test user has an occupation-related tweet, calculate the votes of “student” and “employee”, and go to next. Otherwise, tag the user “undetermined” and stop. The vote of “student” (“employee”) is the number of the tweets whose tags are “student” (“employee”).

4. With the votes of “student” and “employee”, the majority vote is used to get the final tag of the user.

The rule-based ensemble learning: From the error analysis of the whole-tweets-based ensemble learning, we observe that the following two kinds of confusions often occur: “undetermined vs. student”, and “undetermined vs. employee”. Therefore, we attempt to correct the “undetermined” tags outputted from the whole-tweets-based ensemble learning as follows. For a test user, if the tag from the whole-tweets-based ensemble learning is “student” or “employee”, use this tag as the final one. Otherwise, use the tag from the individual-tweet-based ensemble learning as the final one.

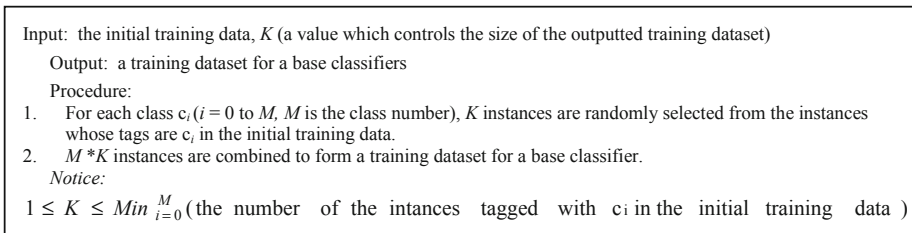


Fig. 1. The class-based random sampling method

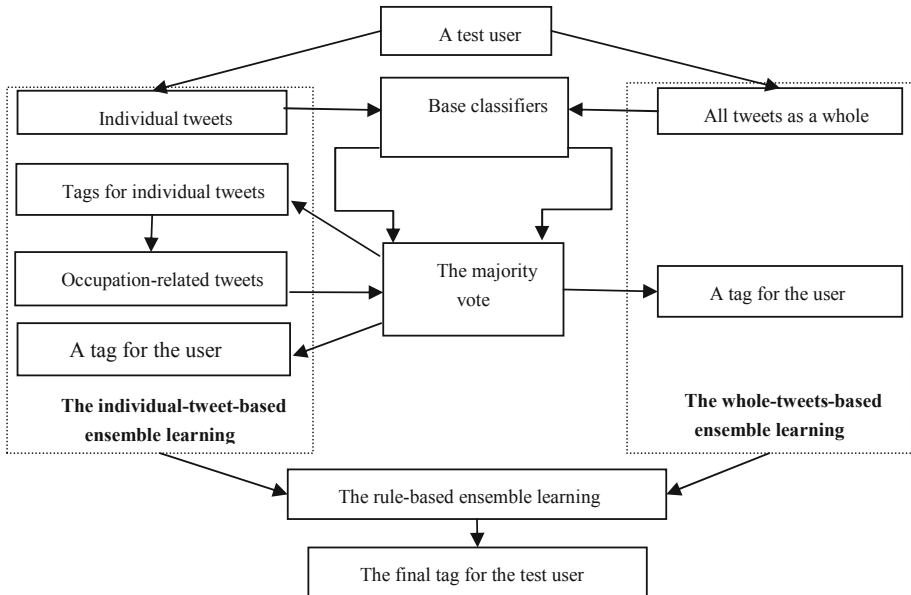


Fig. 2. The cascaded ensemble learning method

5 Experiments

5.1 Experiment Settings

Regarding the experiment data setting, we use the pseudo-training data as the initial training data, the rule-determined test and the rule-undetermined test as the test datasets, and the rule-determined dev and the rule-undetermined dev as the development datasets (see Section 3.3). In addition, in the dev/test data, tag “un-employed” annotated by humans is replaced with tag “undetermined”.

To examine the performances of our MCS-based occupation detection, we implemented two baselines for comparisons. One is a common classification (SC), which uses the following dataset to train one and only one classifier: all of “student” instances (4594 instances), all of “employee” instances (3805 instances) and some of “undetermined” instances (4594 instances). The other baseline (UndSamp+WTEensem) is the under-sampling used in [13]. Similar to our occupation detection, it is also MCS-based and uses random under-sampling and the whole-tweets-based ensemble learning. Moreover, four common measures are chosen for evaluation, i.e. precision (Prec), recall (Rec), F-score (Fs), and accuracy (Acc).

For any supervised classifiers, such as SC and the base classifiers in a MCS-based framework, the Maximum Entropy model implemented by the package Mallet¹ is chosen as the classification method, and the bag of words is used as features.

Regarding the parameter setting in the experiments, all of them are learned from the development datasets. In particular, two parameters, L and K, are very important. L is the number of the base classifiers in a MCS-based framework, and K is the parameter of our class-based random sampling. In our experiments, K is 500 and L is 100.

5.2 The Performances of Different Occupation Detection Models

Table 1 and 2 list the performances of the four occupation detection models for the two test datasets, the rule-determined test (rule-det) and the rule-undetermined test (rule-undet), respectively. To examine our sampling and ensemble learning separately, we develop two MCS-based occupation detection models: RanSamp+WTEensem and RanSamp+CasEnsem. The former uses the class-based random sampling and the whole-tweets-based ensemble learning, and the latter uses the class-based random sampling and the cascaded ensemble learning.

From Table 1 and 2, first, we find that the final model, RanSamp+CasEnsem, significantly outperforms the SC model with 9.9% for “rule-det” and 11.4% for “rule-undet” in F score. This indicates that our class-based random sampling and cascaded ensemble learning work very well.

Second, from SC to UndSamp+WTEensem, a significant improvement is achieved (4.0% for “rule-det” and 4.6% for “rule-undet” in F score). This indicates that a MCS-based framework with a sampling method can effectively overcome the data imbalance and the data bias in our pseudo-training data. Moreover, when the under-sampling (UndSamp+WTEensem) is replaced by our class-based random sampling

¹ <http://mallet.cs.umass.edu/>

Table 1. The performances of the different models for the rule-determined test

	Prec	Rec	Fs	Acc
SC	62.2	65.6	60.7	67.6
UndSamp+WTEensem	64.1	66.8	64.7	73.6
RanSamp+WTEensem	68.6	73.2	69.8	77.0
RanSamp+CasEnsem	69.5	72.6	70.6	77.7

Table 2. The performances of the different models for the rule-undetermined test

	Prec	Rec	Fs	Acc
SC	57.9	54.4	50.3	50.0
UndSamp+WTEensem	60.2	56.1	54.9	54.6
RanSamp+WTEensem	63.3	59.9	58.4	58.2
RanSamp+CasEnsem	63.3	62.8	61.7	61.9

(RanSamp+WTEensem), the performances are further improved (5.1% for “rule-det” and 3.5% for “rule-undet” in F score). This indicates that our class-based random sampling not only can overcome the skewed class distribution problem, but also can effectively reduce the bad effect from the data noise.

Third, from RanSamp+WTEensem to RanSamp+CasEnsem, a significant improvement (3.3% in F score) is achieved for “rule-undet”, and however, a slight improvement (0.8% in F score) for “rule-det”. We observe that the improvement of the RanSamp+CasEnsem model is from the significant improvements of “employee” (4.8% in F score) and “undetermined” (4.2% in F score). This indicates that our individual-tweet-based ensemble learning can effectively solve the confusion of “employee vs. undetermined”. Moreover, from the error analysis for the RanSamp+WTEensem model, we find that this kind of confusions occur much more often in “rule-undet” than in “rule-det”. Therefore, “rule-undet” takes more benefit from our individual-tweet-based ensemble learning than “rule-det” does.

6 Conclusion

In this paper, we make a pilot study for occupation detection for micro-blogging users, and find that even a simple occupation detection task, which detects only the three types of occupation information, is a difficult research issue.

According to the available micro-blogging resources, we proposed a weakly-supervised user occupation detection which achieves a significant improvement. Through the experiments, we realized the main challenges in the user occupation detection, and examine the contributions of different kind of user information to the occupation detection. We believe that the current study should lay ground for future research on occupation detection for micro-blogging users.

Acknowledgement. This work was supported by Key Lab of Information Network Security, Ministry of Public Security.

References

1. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plaintext collections. In: Proceedings of the 5th ACM International Conference on Digital Libraries (2000)
2. Artiles, J., Gonzalo, J., Sekine, S.: WePS 2 Evaluation Campaign: Overview of the Web People Search Attribute Extraction Task. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI (2007)
4. Barandela, R., Sanchez, J., Garcia, V., Rangel, E.: Strategies for Learning in Class Imbalance Problems. *Pattern Recognition* (2003)
5. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* (2002)
6. Chen, Y., Lee, S.Y.M., Huang, C.: PolyUHK: A Robust Information Extraction System for Web Personal Names. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)
7. Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., Sheth, A.: Context and domain knowledge enhanced entity spotting in informal text. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 260–276. Springer, Heidelberg (2009)
8. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Proc. Int'l J. Conf. Intelligent Computing, pp. 878–887 (2005)
9. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: Proc. Int'l J. Conf. Neural Networks, pp.1322–1328 (2008)
10. He, H., Garcia, E.: Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21(9), 1263–1284 (2009)
11. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In: Proc. Int'l Conf. Machine Learning (1997)
12. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, Inc., Hoboken (2004)
13. Li, S., Wang, Z., Zhou, G., Lee, S.Y.M.: Semi-supervised Learning for Imbalanced Sentiment Classification. In: Proceedings of IJCAI (2011)
14. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing Named Entities in Tweets. In: ACL (2011)
15. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under Sampling for Class Imbalance Learning. In: Proc. Int'l Conf. Data Mining, pp. 965–969 (2006)
16. Minkov, E., Wang, R.C., Cohen, W.W.: Extracting personal names from emails: Applying named entity recognition to informal text. In: HLT/EMNLP (2005)
17. Sarawagi, S.: Information Extraction. *Foundations and Trends in Databases* (2008)
18. Wang, B.X., Japkowicz, N.: Imbalanced Data Set Learning with Synthetic Samples. In: Proc. IRIS Machine Learning Workshop (2004)
19. Zhang, J., Mani, I.: KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In: Proc. Int'l Conf. Machine Learning (ICML 2003), Workshop Learning from Imbalanced Data Sets (2003)

Normalization of Chinese Informal Medical Terms Based on Multi-field Indexing

Yunqing Xia¹, Huan Zhao¹, Kaiyu Liu³, and Hualing Zhu²

¹ Department of Computer Science,
TNList, Tsinghua University, Beijing 100084, China
yqxia@mail.tsinghua.edu.cn

² Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong
hzhaoaf@ust.hk

³ Information Networking Institute
Carnegie Mellon University, Pittsburgh, PA 15213, USA
liukaiyu1991@gmail.com

⁴ Asia Gateway Healthcare Information Technology (Beijing) Co Ltd
Beijing 100027, China
hualing.zhu@aghit.com

Abstract. Healthcare data mining and business intelligence are attracting huge industry interest in recent years. Engineers encounter a bottleneck when applying data mining tools to textual healthcare records. Many medical terms in the healthcare records are different from the standard form, which are referred to as informal medical terms in this work. Study indicates that in Chinese healthcare records, a majority of the informal terms are abbreviations or typos. In this work, a multi-field indexing approach is proposed, which accomplishes the term normalization task with information retrieval algorithm with four level indices: word, character, pinyin and its initial. Experimental results show that the proposed approach is advantageous over the state-of-the-art approaches.

Keyword: Term normalization, medical terms, indexing, ranking.

1 Introduction

Healthcare data mining and business intelligence are attracting huge industry interest in recent years. Healthcare industry is benefited from data mining applications for various parties [1] by feeding the demand of efficient analytical methodology for detecting unknown and valuable information in health data. With data mining tools, fraud in health insurance can be detected, medical advices can be provided to the patients at lower cost. Moreover, it becomes possible with data mining tools to detect causes of diseases and identify novel medical treatment methods. On the other hand, healthcare researchers may make full use of business intelligence tool in making efficient healthcare policies, constructing drug recommendation systems, or developing health profiles of individuals.

Engineers encounter a bottleneck when applying data mining tools on textual healthcare records. Many medical terms in the healthcare records are different

from the standard form, which are referred to as informal medical terms in this work. We first give some examples in Table 1.

Table 1. Examples of informal medical terms^p

#	Informal term	Standard Counterpart	English explanation
E1	上感	上呼吸道感染	upper respiratory tract infection
E2	TNB	糖尿病	diabetes
E3	GXB	冠状动脉硬化性心脏病	coronary arteriosclerotic cardiopathy
E4	Guillian-Barre氏综合征	吉兰-巴雷综合征	Guillian-Barre syndrome
E5	急性阑尾炎	急性阑尾炎	acute appendicitis

The examples in Table 1 actually presents the following three categories of informal medical terms:

- Abbreviation: The abbreviations can be further classified into Chinese word abbreviation, pinyin abbreviation and mixed abbreviation. For example, *E1* (‘上感’, *shang4 gan3*) is a Chinese word abbreviation, *E2* (TNB) is pinyin abbreviation and *E3* (GXB) is a pinyin abbreviation for ‘冠心病(*guan4 xin1 bing4*)’, which is a Chinese word abbreviation for ‘冠状动脉硬化性心脏病(*guan4 zhuang4 dong4 mai4 ying4 hua4 xing4 xin1 zang4 bing4*)’.
- Transliteration: The standard term is a transliteration of a word. In example *E4p*, ‘吉兰-巴雷(*ji2 lan2 - ba1 lei2*)’ is transliteration of *Guillian-Barre*.
- Character input error: Some characters are wrong but phonetically equal / similar to the standard character. In example *E5*, ‘阑(*lan2*)’ is replaced by 烂(*lan4*). This is typically caused by Chinese pinyin input tool.

Study indicates that in Chinese healthcare records, a majority of the informal terms are abbreviations or typos. Thus, targeting at the two types of informal medical terms, we propose a multi-field indexing approach, which is able to normalize the informal medical terms with under the information retrieval framework with four level indices: word, character, pinyin and initial.

The standard medical terms are first segmented using the standard Chinese lexicon, namely, no medical terms is included. For example, term ‘上呼吸道感染(*upper respiratory tract infection, shang4 hu1 xi1 dao4 gan3 ran3*)’ is split into {上呼吸道(*upper respiratory tract*)|感染(*infection*)}. As many abbreviations are generated based on word, we detect boundary of the words for the purpose to discover the word level abbreviations, e.g., *E1* in Table 1. With the fine-grained words, we are also able to handle input errors as well as synonyms, e.g., ‘症(*symptom, zheng4*)’ and ‘病(*textitdisease, zheng4*)’.

We also handle the medical terms on character level. For example, we split ‘糖尿病(*diabetes, tang2 niao4 bing4*)’ into {糖|尿|病}. This makes it possible that we recognize TNB as its abbreviation. Again, the character level treatment is also useful to handle input errors, e.g., *E5* in Table 1.

We further handle the medical terms on pinyin level. Every Chinese character holds a pinyin, which indicates how the character is produced. The purpose is

to recognize the abbreviations that are comprised of initials, e.g., *E2* in Table 1, or English word, e.g. *E5* in Table 1.

We adopt an information retrieval framework in medical term normalization. We first index the words, characters, pinyin's and their initials with Lucene¹. Using the input informal terms as a query, we then apply standard *BM25* algorithm² to retrieve and rank the standard terms in multiple fields. Experimental results show that the proposed approach is advantageous over the state-of-the-art approaches.

The reminder of this paper is organized as follows. The related work is summarized in Section 2. The proposed method is described in Section 3. We present the evaluation as well as discussion in Section 4. We finally conclude this paper in Section 5.

2 Related Work

This work is related to two categories of previous work: medical term normalization and language normalization.

2.1 Medical/Biological Term Normalization

Medical term normalization is a major task in a few bio-medial natural language processing competitions, e.g., 2013 ShARe/CLEF eHealth Shared Task (Disorder Normalization in Clinical Notes) [2] and NTCIR11 Medical NLP Shared Task³ (an ongoing task). The former task focuses on English terms while the latter on Japanese. There is no report so far on other languages. Unified Medical Language System (UMLS) is a commonly used English medical knowledge base [3, 4] which contains over 2 million names for some 900,000 concepts from more than 60 families of biomedical vocabularies, and includes 12 million relations among these concepts. Most recent work on English medical terms are based on UMLS [5, 2].

In the ShARe/CLEF eHealth Shared Task, acronyms/abbreviations are expected to be mapped to standard terms in UMLS. Researchers have developed systems to normalize acronyms/abbreviations in clinical texts for information extraction, information retrieval, and document summarization applications. Wu et al. (2012) compare performance of some current medical term normalization tools, e.g., MetaMap, MedLEE, and cTAKES. It is showed that f-scores of these tools range from 0.03 to 0.73 [6].

As showed in the aforementioned work, lexicon based matching is a dominant solution. In this work, we base our work on the Chinese medical terminology system developed by Asia Gateway Co. Ltd. and explore advanced technology for term matching.

¹ <http://lucene.apache.org/>

² http://en.wikipedia.org/wiki/Okapi_BM25

³ <http://mednlp.jp/ntcir11/#task>

2.2 Language Normalization

Language normalization is an important task for many natural language processing systems such as machine translation, information extraction and information retrieval. The problem is defined as detecting the so-called informal words from text and mapping them to their counterparts in the standard lexicon. In [7], Sproat et al. (2001) propose a ngram language model for this purpose. In [8], Xia et al. (2006) propose a phonetic model for chat language normalization.

The common characteristics of the above research lies in that they handle free natural language text. This work is different because we handle medical terms which cannot be directly mapped to any term in the standard medical lexicon. Our input is not the free text, but the terms that are already detected as non-standard ones.

3 Method

3.1 The Workflow

We adopt the information retrieval framework and accomplish the medical term normalization task via term retrieval and ranking. The general workflow is given in Fig 1.

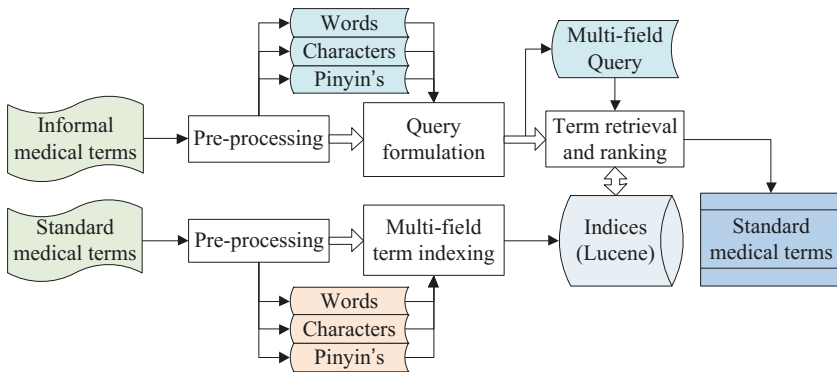


Fig. 1. The general workflow for medical term normalization.

The input to the system is medical terms which cannot be matched in the current ontology. The output is the standard counterpart terms. The horizontal part of the workflow is designed for standard term indexing, in which standard terms are first converted into sequences of words, characters and pinyin's, respectively. The words, characters and pinyin's are in turn input to Lucene to build a

multi-filed index. The vertical part of the workflow is designed for informal term normalization. Based on standard *BM25* score, we are able to retrieve and rank the relevant standard terms. The workflow is elaborated as follows.

3.2 Multi-field Term Indexing

Observation shows that informal medical terms are created with variations in word, character and pinyin levels, no matter deliberately or not. Enlightened by this, we propose to create index for the standard medical terms with multiple fields so that the given informal term can be matched within the standard terms according to relevance in word, character and pinyin levels.

Applying ICTCLAS⁴, we first segment the input informal term in to words. Note that only standard vocabulary is used in ICTCLAS, thus a medical term is usually segmented into a few word. For example, '冠状动脉硬化性心脏病(*guan4 zhuang4 dong4 mai4 ying4 hua4 xing4 xin1 zang4 bing4*)' is segmented into {冠状动脉(*coronary artery*)|硬化(*sclerosis*)|性(*type*)|心脏病(*heart disease*)}.

According to unicode scheme, i.e. UTF-8, for the Chinese characters, we further split words into characters. For example, the above medical term is segmented into {冠|状|动|脉|硬|化|性|心|脏|病}.

We further apply pinyin annotator, i.e. pinyin4j⁵, to transcribe every Chinese word with pinyin. For example, pinyin annotation of '冠状动脉' is *guan4 zhuang4 dong4 mai4*. Note that the tool works better when a whole word is input because there are some Chinese characters which can be mapped top different pinyin's in various contexts.

There are some cases that English words/characters are contained in the Chinese medical terms. We use natural delimiter to split the English word/s/characters. For example, 'Guillian-Barre氏综合征{*Guillian-Barre syndrome, Guillian-Barre shi4 zong1 he2 zheng4*}

 in example *E4* in Table 1 is split to {Guillian-|Barre|氏|综|合|征}.

The obtained words, Chinese characters and pinyin's are used to create a multi-filed index with the following structure:

```
INDEX = {
    string Words;\\Words in the term to be indexed
    string WordInitials;\\The first characters of the words
    string Pinyins;\\Pinyin's of the term
    string PinyinInitials;\\Initials of the above pinyin's
    string PinyinFinals;\\Finals of the above pinyin's
    string Characters;\\Characters of the term
}
```

In a visual manner, the index for the term '上呼吸道感染{*upper respiratory tract infection, shang4 hu1 xi1 dao4 gan3 ran3*}

 is indexed as follows:

⁴ <http://ictclas.org/>

⁵ <http://pinyin4j.sourceforge.net/>

```

INDEX = { \\ '上呼吸道感染'
  [上呼吸道 感染] \\Words
  [上感] \\WordInitials
  [shang4 hu1 xi1 dao4 gan3 ran3] \\Pinyins
  [s h x d g r];\\ PinyinInitials
  [ang u i ao an an] \\ PinyinFinals
  [上呼 吸 道 感 染] \\Characters
}

```

Note in the INDEX structure that *Word initials*, *Pinyin initials*, *Pinyin finals* and *non-Chinese characters* are involved. They are considered as fields in the index due to the following reasons:

- *Word initials*: Some informal terms are created using abbreviation of word initials, e.g., *E1* in Table 1.
- *Pinyin initials*: Some informal terms are created using abbreviation of pinyin initials, e.g., *E2* in Table 1.
- *Pinyin finals*: Some informal terms are created due to an error input, e.g., '阿司品林(*a1 si1 pin3 lin2*)' which corresponds to '阿司匹林(*asprin, a1 si1 pi1 lin2*)'.

In this work, all the medical terms in the Chinese medical ontology are dumped into the Lucene system.

3.3 Term Retrieval and Ranking

As standard medical terms are indexed in multiple fields, the informal term input to the Lucene system should also be pre-processed in the same way so as to match the standard terms in the according fields. As showed in Fig 1, the words, characters and pinin's are first used to form a multi-filed query. Using the informal term '上感(*shang4 gan3*)' in example *E1* as example, the query is formed as follows:

```

INDEX = { \\ '上感'
  Words = [上感]
  WordInitials = [上感]
  Pinyins = [shang4 gan3]
  PinyinInitials = [s g]
  PinyinFinals = [ang an]
  Characters = [上感]
}

```

The *BM25* algorithm⁶ is a standard relevance calculation algorithm. Note that the *BM25* algorithm assigns the multiple fields equal weights. However, our empirical study indicates that the weights vary significantly considering their

⁶ http://en.wikipedia.org/wiki/Okapi_BM25

Table 2. Empirical weights of the multiple fields in the index

Field	Weight
Words	1
WordInitials	0.1
Pinyins	3
PinyinInitials	0.1
PinyinFinals	2
Characters	1

contribution to term retrieval. Using 100 human judged pairs of informal terms and their standard counterparts, we obtain the weights for the fields in Table 2.

At last, the *BM25* algorithm output top N standard terms which hold biggest relevance score.

4 Evaluation

4.1 Setup

Dataset

In this work, 300 pairs of informal medical terms and their standard counterparts are compiled by medical experts. The dataset covers 125 Chinese abbreviations, 48 pinyin abbreviations and 127 typos.

In our medical ontology, 48,000 medical terms are covered, which are used as standard terms.

Evaluation Metric

We first adopt precision in top N terms (i.e. $p@N$) as evaluation metric. That is, we calculate percentage of correctly normalized terms amongst all the input informal terms. Meanwhile, we adopt execution time to compare computational complexity of the methods.

4.2 Experiment 1: Normalization Methods

In this experiment, we intend to compare our proposed method against the following two baseline methods:

- *Edit distance* (EDDis): Similarity of the input informal terms and standard term is calculated using edit distance⁷ in the six fields in Section 3.3. The wights in Table 2 are used to combine the six similarity values in a linear manner, thus an overall similarity value is obtained for the input term and the standard term.
- *Multi-filed cosine similarity* (MSim): This method differs from the edit distance method in calculating similarity with cosine formula.

⁷ http://en.wikipedia.org/wiki/Edit_distance

In the proposed IR-based normalization method (IRNorm), the retrieval and ranking process are delivered with *BM25* within Lucene. Experimental results are presented in Table 3.

Table 3. Experimental of different methods for medical term normalization

Method	p@5	p@10	time (milliseconds per term)
EDDis	0.748	0.762	120
MSim	0.853	0.892	180
IRNorm	0.892	0.907	6*

* The total time for indexing the 48,000 standard terms is 30 seconds.

Discussion

It can be seen from Table 3 that the precision values of MSim and IRNorm are close, while are both higher than EDDis in both top 5 and top 10 terms. This indicate that cosine distance and *BM25* are both more effective than edit distance in term normalization.

We also notice that EDDis and MSim require much longer computing time. Counting in the indexing time, i.e. 1620 seconds in total, the proposed IRNorm method is much faster than EDDis and MSim. This ascribes to the Lucene engine.

4.3 Experiment 2: The Fields in the Index

In this experiment, we seek to evaluate contribution of the fields in the index. Using different configuration, we develop various implementations of the proposed method, showed in Table 4.

Table 4. Implementations of our method with different configuration

Method ID	Word	Character	Pinyin
IRNorm-A	Y	N	N
IRNorm-B	Y	Y	N
IRNorm-C	Y	N	Y
IRNorm-D*	Y	Y	Y

* This is the method used in Experiment 1.

Experimental results are presented in Table 5.

Discussion

It can be seen from Table 5 that IRNorm-B outperforms IRNorm-A by 0.226 on p@5 and by 0.241 on p@10. This indicates that the Chinese character makes significant contribution to term normalization. Comparing IRNorm-A and IRNorm-C, we find pinyin makes more contribution, i.e., improving by 0.297 on p@5 and by 0.311 on p@10. When the Chinese character and pinyin are both

Table 5. Experimental results (p@N) of different implementations of our method

Implementation	p@5	p@10
IRNorm-A	0.398	0.412
IRNorm-B	0.624	0.653
IRNorm-C	0.685	0.723
IRNorm-D	0.892	0.907

used, the outperformance is significant. This indicates that the fields that are defined in this work are indeed helpful in discovering the standard counterparts of the input informal terms.

5 Conclusion and Future Work

The informal medical terms pose huge challenge to medical business intelligence systems. In this work, the term normalization task is accomplished with an information retrieval framework using multi-field index. The contributions of this work are summarized as follows. Firstly, the proposed method considers words, Chinese characters and pinyin's in standard term matching, which makes term normalization more accurate. Secondly, with the multi-field index under information retrieval framework, the normalization process is made much faster. Experiments show the proposed method is very effective.

Note that this work is still preliminary. The following work is planned. First, only intra-term features are considered in term normalization. Therefore, we will explore how context helps to normalize the informal medical terms. Secondly, in our term normalization system, the input is informal term, which is detected by human experts. However, in many cases the informal term should be detected automatically by machines. Thus in the future we will develop the informal term detection algorithm.

Acknowledgement. This work is supported by National Science Foundation of China (NSFC: 61272233). We thank the anonymous reviewers for the valuable comments.

References

1. Koh, H., Tan, G.: Data mining applications in healthcare. *J. Healthcare Inf. Manag.* 19(2), 64–72 (2005)
2. Suominen, H., et al.: Overview of the shARe/CLEF eHealth evaluation lab 2013. In: Forner, P., Müller, H., Paredes, R., Rosso, P., Stein, B. (eds.) *CLEF 2013*. LNCS, vol. 8138, pp. 212–231. Springer, Heidelberg (2013)

3. Campbell, K.E., Oliver, D.E., Shortliffe, E.H.: The unified medical language system: Toward a collaborative approach for solving terminologic problems. *JAMIA* 5(1), 12–16 (1998)
4. Bodenreider, O.: The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research* 32(database issue), 267–270 (2004)
5. Kim, M.Y., Goebel, R.: Detection and normalization of medical terms using domain-specific term frequency and adaptive ranking. In: 2010 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB), pp. 1–5. IEEE (2010)
6. Wu, Y., Denny, J., Rosenbloom, S., Miller, R., Giuse, D., Xu, H.: A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries. In: *AMIA Annu. Symp.*, 997–1003 (2012)
7. Sproat, R., Black, A.W., Chen, S.F., Kumar, S., Ostendorf, M., Richards, C.: Normalization of non-standard words. *Computer Speech & Language* 15(3), 287–333 (2001)
8. Xia, Y., Wong, K.F., Li, W.: A phonetic-based approach to chinese chat text normalization. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 993–1000. Association for Computational Linguistics, Stroudsburg (2006)

Answer Extraction with Multiple Extraction Engines for Web-Based Question Answering

Hong Sun¹, Furu Wei², and Ming Zhou²

¹ School of Computer Science and Technology, Tianjin University, Tianjin, China
kaspersky@tju.edu.cn

² Microsoft Research Asia, Beijing, China
{fuwei, mingzhou}@microsoft.com

Abstract. Answer Extraction of Web-based Question Answering aims to extract answers from snippets retrieved by search engines. Search results contain lots of noisy and incomplete texts, thus the task becomes more challenging comparing with traditional answer extraction upon offline corpus. In this paper we discuss the important role of employing multiple extraction engines for Web-based Question Answering. Aggregating multiple engines could ease the negative effect from the noisy search results on single method. We adopt a Pruned Rank Aggregation method which performs pruning while aggregating candidate lists provided by multiple engines. It fully leverages redundancies within and across each list for reducing noises in candidate list without hurting answer recall. In addition, we rank the aggregated list with a Learning to Rank framework with similarity, redundancy, quality and search features. Experiment results on TREC data show that our method is effective for reducing noises in candidate list, and greatly helps to improve answer ranking results. Our method outperforms state-of-the-art answer extraction method, and is sufficient in dealing with the noisy search snippets for Web-based QA.

Keywords: Web-based Question Answering, Answer Extraction, Rank Aggregation, Learning to Rank.

1 Introduction

Question Answering (QA) aims to give exact answer to questions described in natural language. Some QA systems directly employ well-built search engines for this task which are called Web-based QA systems [1]. This kind of systems contain three modules: 1) question analysis module to analyze question and generate queries; 2) passage retrieval module to retrieve relevant passages from search engine; 3) answer extraction module to extract the final answer. Comparing with traditional QA, Web-based QA can take advantage of the tremendous data resource provided by Web and eliminate the efforts to store and index huge amount of documents. Current search engines are becoming more and more sophisticated, so Web-based QA can also benefit from the optimized search results provided by search engines.

We focus on Answer Extraction task for Web-based QA. The task is to generate exact answers with search snippets. It contains two steps: extract candidates such as noun phrases and later rank them based on ranking function, e.g., similarity between question and sentence bearing the candidate. Traditional web-based answer extraction is conducted on search snippets [1] instead of plain texts in the retrieved web pages, so that it can utilize the high-quality summarizations generated by search engines without parsing or mapping sentences into the original web pages. The side-effect is that answer extraction results are influenced by the incomplete and noisy search snippets. On the one hand, using search snippets containing many negative sentences results with large amount of noisy candidates; on the other, state-of-the-art answer extraction methods rely on syntactic information [2,3] could be seriously affected by the incomplete structure of search text. Previous work about Web-based QA have discussed using n-grams as candidates and rank them based on redundancies [10]. This method eliminates the need of deep understanding of the noisy search snippets and leverages substantial redundancy encoded in the large amount of search texts. However, enumerating n-grams results with even more negative candidates in the hypothesis space which poses too much address on ranking [4].

Single extraction method is easily affected by the noisy search snippets, in order to ease the problem, we discuss an effective way by adopting multiple extraction engines for Web-based QA. Different engines analyze texts from different aspects, the chance for them all being wrong is small, thus the consensus information among them is useful for alleviating the impact from the noisy texts. With multiple engines, we can perform more strict pruning to filter noisy candidates and perform more accurate ranking. Specifically, we first aggregate multiple extraction engines' results with a Pruned Rank Aggregation method. We employ a modified Supervised Kemeny Ranking model for rank aggregation and at the same time perform pruning on each engine's list based on redundancies within and across different extraction engines. After generating the list, we use Learning to Rank with similarity, redundancy, quality and search features to rank the candidate list. Experiments on TREC dataset show that our pruning and ranking algorithm is efficient for reducing noises in candidate list and achieving better ranking results than state-of-the-art answer extraction method. This result makes Web-based QA more accurate, robust and applicable in real application scenarios.

2 Related Work

Our work focus on Answer Extraction of Web-based QA. Methods for this task can be classified as two types.

The first type of methods consider information from question when generating candidates and use relatively simple ranking functions. For example, in traditional QA, one most commonly adopted method is to extract Named Entities (NE) matching with answer type with Named Entity Recognition (NER) tools [5]. For this method, errors in answer type classification will propagate to

the stage of answer extraction, and performance of answer extraction will be limited by performance of NER. In Web-based QA, search snippets are different from training texts of NER, this makes results of NER significantly degraded [4]. Pattern-based method is another commonly used method [6]. It has high precision, but those patterns are defined on predicates which are very fine-grained and not easy to be adapted to new data. Recent work has studied to apply machine learning in answer extraction. The first attempt is to view answer extraction as a process to extract question-biased terms [7], each token in passage is classified as is or is not an answer token. Within this direction, factor graph [4] is used to consider the relations among different passages, and Tree Edit Distance is considered in later work [2] for a more accurate similarity measurement between question and passages.

The second type of method is to generate candidates based on text's structure or dictionaries. Noun Phrase (NP) is one commonly used unit [3], and ranking score of an NP can be calculated based on syntactic structure of passage. In the state-of-the-art method [3], each NP considered as a candidate is aligned to Focus [8] on the parsing tree; thus for each alignment, a similarity score could be calculated based on tree kernel. This method relies on syntactic structure and is affected by the noisy search snippets. We'll show in our experiments that the performance of this method seriously degraded in the search contexts of Web-based QA. Besides, some work use dictionaries to generate candidates, for example, state-of-the-art QA system Watson [9] extracts Wikipedia titles from passages, and employs Learning to Rank with hundreds of features in answer ranking. This method requests lots of human effort for feature engineering has low adaptability to new domains [3]. Besides, n-grams in texts are commonly employed in Web-based QA [10]. Such method uses answer type to filter out candidates and perform tiling on candidates with overlaps. Frequencies weighted with prior score of query are viewed as the ranking score. Extracting n-grams results with large amount of candidates containing more noises thus requires more sophisticated ranking function and features [4].

3 Method

Our method for answer extraction contains two steps: 1) joint aggregation and pruning of candidate lists generated by different extraction engines; 2) rank the candidate list generated by previous step.

3.1 Pruned Rank Aggregation

The first step could be viewed as a rank aggregation task of candidate lists generated by multiple extraction engines. As search snippets provide various of texts from Web, they contain many noises. For example, in our statistics 19% sentences in TREC corpus [2] contain correct answer, while such ratio is around 10% in search result. This indicates more negative snippets without containing correct answer will bring many noisy candidates to answer extraction. Thus it's

necessary to perform strict pruning during or after the aggregation to reduce the amount of noises.

Algorithm 1. Pruned Rank Aggregation with Multiple Extraction Engines

Input: $C_i = \{c_{i1}, \dots, c_{il}\}$ as ordered candidate list from engine i , $l_i = |C_i|$, each list ordered by frequency of candidate $n_{c_{ij}}$; $\{w_1, \dots, w_k\}$ are weights of each engine, $0 \leq w_i \leq 1, \sum_i w_i = 1$; $\{t_i, \dots, t_k\}$ are single engine pruning thresholds, $0 \leq t_i \leq 1$; p, m, n are pruning thresholds after aggregation.

Output: $A = a_1, \dots, a_n$ is candidate list after aggregation and pruning

- 1: Initialize $A = \emptyset$
- 2: **for all** C_i **do**
- 3: **for all** $c_{ij} \in C_i$ **do**
- 4: **if** !ContainContentWord(c_{ij}) or ($j > t_i \cdot l_i$ and !Exists($c_{ij}, C'_i, \forall i' \neq i$))
 then Remove c_{ij} from C_i
- 5: **end if**
- 6: **end for**
- 7: Update $l_i = |C_i|$
- 8: **end for**
- 9: **function** MERGELISTWITHMODIFIEDSKR($\{C_i\}$)
- 10: Initialize $M_{i,j} \leftarrow 0, C' = \bigcup_i C_i$, update frequency $n_{c_k} = \sum_j n_{c_{ij}}$, if $c_k = c_{ij}$
- 11: **for all** candidate list C_i **do**
- 12: **for all** $j = 1$ to $l_i - 1$ **do**
- 13: **for all** $l = j + 1$ to l_i **do** $M_{c_{ij}, c_{il}} \leftarrow M_{c_{ij}, c_{il}} + w_i \cdot \log(n_{c_{ij}} - n_{c_{il}} + 0.1)$
- 14: **end for**
- 15: **end for**
- 16: **end for**
- 17: Quick sort C' with $M_{C'_i, C'_j}$, candidate with larger value gets prior order
- 18: **return** C'
- 19: **end function**
- 20: Compute word frequency f_{w_j} for all $w_j \in c_i$ and !IsStopword(w_j), where $c_i \in C'$
- 21: **for all** $c_i \in C'$ **do**
- 22: **if** $i \leq |C'| \cdot p$ or $\Pi f_{w_i} \geq n, w_i \in c_i$ or $n_{c_i} \geq m$ **then** Add c_i to A
- 23: **end if**
- 24: **end for**

Thus we propose a Pruned Rank Aggregation method for joint aggregating and pruning candidate lists generated by multiple extraction engines. In search results, texts are more redundant than off-line corpus, so redundancy provides us with useful information in distinguishing between good and low-quality candidates. In addition, as different extraction engines perform analysis differently, if a candidate is supported by multiple engines, it is unlikely that it's a noisy candidate. Thus we design the method for pruning based on inner and inter redundancies of different extraction engines during rank aggregation. The algorithm is shown in Algorithm 1. Given different candidate lists pre-ordered by frequencies of candidates, line 1-8 first prune each candidate list to filter candidates existing in only one list and with low rank¹. Line 9-19 employ a modified

Supervised Kemeny Ranking (SKR) [11] to aggregate different lists, element of matrix in SKR is weighted by difference between frequencies of two candidates in the same list (line 13). Later, the aggregated list is further pruned at line 20-24 based on global candidate-based and word-based frequencies. Pruning parameters $\{t_i\}$, p, m, n in the algorithm are tuned with development set; w_i is calculated as accuracy of each engine in development set. After Pruned Rank Aggregation, we get a candidate list with less noisy candidates, and at the same time, recall of answer is not hurt comparing to aggregation without pruning. Such a result helps to increase answer ranking result in the later stage.

3.2 Ranking

Ranking task is that given question Q , search snippets $S = \{s_1, s_2, \dots, s_n\}$, external Knowledge Base K , candidates $A = \{a_1, a_2, \dots, a_m\}$, perform ranking with scoring function:

$$\begin{aligned} f(a_i) &= P(a_i | Q, S, K, A) \\ &= \sum_j \lambda_j \cdot h_j(a_i, Q, S, K, A) \end{aligned} \quad (1)$$

where $\{h_j(\cdot)\}$ is a set of ranking features and λ_j is the corresponding feature weight. Evidences for ranking in this work come from question, search results, Knowledge Base as well as the whole hypothesis space.

After generating the score, answer extraction ranks candidates with the score and selects candidate answer with the highest score as the final output:

$$\hat{a} = \arg \max_{a_i \in A} f(a_i) \quad (2)$$

Previous work has showed that Learning to Rank works well for answer ranking of factoid question [12]. In this work, we follow the strategy and adopt *Rank SVM* [13] as our answer ranking model. It converts ranking problem to a binary classification problem during training, each pair of candidates is viewed as a positive training sample if correct candidate's score is higher than negative ones' and vice versa. During predication, each candidate's score is estimated with features and the weights.

We define 5 different feature sets to capture quality of a candidate from different aspects(number in () indicates number of the features):

Consensus Features (9). This set measures candidates' agreement across different extraction engines. Such feature set includes: number of engines generating this candidate; variance of frequencies of the three engines; reciprocal rank of the candidate in each engine's ranking list (pre-ranked by the same ranking model in this work without consensus features); score of candidate in each engine's list; number of occurrences identified by two/three engines at the same time.

¹ We collect stopword list from <http://www.ranks.nl/stopwords>. Besides those stopwords, all words are viewed as content words.

Redundancy Features (7). This set measures redundancy for each candidate in the whole hypothesis space and includes: frequency of the candidate; number of different search snippets and passages containing the candidate; n-gram based redundancy score:

$$f(a_j, A) = \sum_i^T p(tk_i) \quad (3)$$

where tk_i is n-gram in a_j , T is the number of the n-grams. Score of an n-gram is estimated with:

$$p(tk_i) = \sum_{j=1}^N \frac{n_{a_j} * \delta'(tk_i, a_j)}{n_j} \quad (4)$$

where $\delta'(tk_i, a_j)$ is the indicator function equals to 1 if tk_i appears in candidate a_j and 0 otherwise, n_{a_j} is the frequency of a_j ; N is the number of candidates, n_j is the number of n-grams in a_j . We compute the above scores of unigram and bigram, in addition with such scores normalized by candidate's length.

Similarity Features (41). This set measures similarity between candidate and question, including text similarity of candidate's context and semantic similarity between candidate and answer type. For text similarity, to avoid syntactic parsing we use term level similarities such as: Longest Common Sequence (LCS), sequential LCS, edit distance, number overlapped content words in sentences, and number of overlapped content words in contexts around candidate and question focus (phrase in question could be replaced by answer [14]) respectively. Those similarities are calculated both in passage level and sentence level (each search snippet contains about 3 sentences separated by "..."), and normalized by question length and sentence/passage length respectively. In addition, 9 similarity features are based on word embedding [15] such as average similarity between question phrases and search snippet phrases.

For semantic similarity, we measure whether the candidate matches with answer type or Lexical Answer Type (LAT) [14]. LAT is more specific concept than answer type, such as *Doctor*, *city*, etc. We build answer type dictionary from FreeBase ² and adopt NeedleSeek [16] as LAT dictionary. We also build regular expressions to identify quality and date candidates.

Candidate Quality Features (9). This set measures the candidate's own quality, including: whether the candidate is capitalized; number of content words in the candidate and the value normalized by total token number of the candidate; number of candidate's tokens in question and the value normalized by total token number; length of the candidate.

Search Features (7). This set includes average, worse and best rank of snippets bearing the candidate given by search engine; the candidate is extracted from search snippets or titles; whether the candidate is extracted from Wikipedia site.

² www.freebase.com

3.3 Extraction Engines

In this work we employ three extraction engines:

Sequential Labeling Method (CRF). The first method is sequential labeling method similar with previous work [2,4,7]. Each token in passage is labeled as is or is not an answer token. Question-answer pairs are used to generate training data for this method. Given a question Q , answer A and search snippets S , sequential tokens in S are labeled with 1 if they match with A and 0 otherwise. We adopt 15 features similar to previous work [4] but without syntactic-based features, and employ CRF [17] as our labeling model. In order to increase recall of this method, we follow the setting of forced CRF [2], that for each passage, beside tokens with positive labels, we change top k (k is set to 2 based on results in development set) negative tokens' labels to positive (those tokens are ranked based on their scores of positive label). If continuous tokens' labels are 1, they are merged to form a single candidate answer.

Wikipedia Title-Based Method (Wiki). Following state-of-the-art system's method [9], we extract Wikipedia title entries appeared in search snippets. Specifically, we build a dictionary of 7.8 million entries consisting of all Wikipedia title entries. Given a search snippet, we scan the snippet with forward maximum matching and extract all the matched entries case-insensitively.

Noun Phrase-Based Method (NP). As we focus on factoid questions, Noun Phrases cover most of the correct answers. Noun phrases are extracted from search snippets, we adopt Stanford parser ³ to identify noun phrases.

4 Experiments

4.1 Experiment Setup

Our data is collected from TREC data. We build training and development set from TREC. There are two test sets in our work. The first one (Test-1) is the one used in previous work [2,3], previous work have provided documents for answer extraction for this set. The second one (Test-2) is a larger test set, but there's no documents provided for this set. For all the training, development and testing sets, we collect search snippets for each question. The snippets are retrieved by five queries: the question; verb, noun, adj, adv words in question; Named Entities (if any) in question; noun phrase and verbs; verb and its dependents in question. For each query we collect top 20 snippets returned by search engine, and select 60 search snippets most similar to the question from them. Summary of the data is shown in Table 1. *Avg. Passage per Q* indicates in average for each question, there are how many passages available for answer extraction; *Rate of Positive Passages Per Q (%)* indicates among all the passages, the rate of positive passages that contains the correct answer.

Table 1. Answer extraction training and testing data used in this work

Data	Questions	Passage Type	Avg. Passage Per Q	Rate of Positive Passages Per Q (%)
Train	1200	Search	60	9.82
Test-1	75	Search	60	10.33
	89	Document	17	18.72
Test-2	293	Search	60	10.11

Table 2. Comparative results on testing set

Testset	Passage	Method	Top 1 Acc.	Top 5 Acc.	MRR
Test-1	Document	Tree Kernel	70.79	82.02	73.91
		Our Method	69.66	79.78	72.12
Test-1	Search	Tree Kernel	52.00	78.67	58.17
		Our Method	66.67	84.00	70.71
Test-2	Search	Tree Kernel	51.19	72.35	59.81
		N-gram	50.85	72.70	60.78
		Our Method	66.55	79.52	69.93

Our CRF-based answer extraction method and feature weights of ranking model is trained on our training set with search texts. Parameters of pruning are tuned with development set, $w_{crf} = 0.34$, $w_{wiki} = 0.33$, $w_{np} = 0.33$, $t_{crf} = 0.9$, $t_{wiki} = 0.75$, $t_{np} = 0.9$, $p = 0.9$, $m = 2$ and $n = 2$. We compare our method with state-of-the-art answer extraction method using Tree Kernel[3]. We re-implement the method and perform training and testing on our dataset. The author didn't mention which chunker or parser they employ, so we use Stanford parser to generate parsing trees and extract NP chunks from the trees.

We report Top 1 Accuracy (ratio of correctly answered questions with the first candidate in the ranking list), Top 5 Accuracy (ratio of correctly answered questions with the first 5 candidates) and MRR (multiplicative inverse of the rank of the first correct answer) metrics.

4.2 Experiment Results

Compare to State-of-the-art Method

The comparative result with Tree Kernel method is shown in Table 2⁴. In document set, Tree Kernel based method slightly outperforms ours. When adapting both methods from regular document to search text, they all degrade. In search set, Tree Kernel method's performance seriously drops from 71% accuracy to 52%, while in contrast, our method is less affected and outperforms Tree Kernel

³ <http://www-nlp.stanford.edu/software/>

method on search texts. This indicates our method is more efficient for Web-based answer extraction.

For both methods, results on document sets are superior to the one on search sets, this is because search snippets contain more noises than documents. As it's shown in Table 1, rate of positive passages containing the correct answer in document set is 18.72%, while in contrast, such rate is much lower in search set. So both of the answer extraction methods have to deal with more negative candidates in the hypothesis space. Specifically, positive and negative candidates' rate of our method is 1: 27 in document set, while such rate is 1: 59 in search set before pruning. In addition, search snippets are often incomplete, and sometimes key words in question are in sub-sentence apart from the correct answer. We observe that about half of the positive sentences containing the correct answer don't contain any key words in question or aren't complete sentences. Under such condition, syntactic similarity between question and answer bearing sentences will be reduced. The result is that traditional method such as Tree Kernel method relies on such similarity will degrade on the ill-formed search snippets. Previous work [10] discuss to use n-gram for answer extraction, as a result, number of negative samples in the hypothesis space of that method will also be larger. As Table 2 shows, the n-gram based method also performs worse than our method.

Single Extraction Versus Multiple Extractions

Using multiple extraction engines brings great contributions. There are two positive effects of the multiple extraction engines-based method: 1) redundancies among different engines enables us to perform more restrict pruning which reduces noises in candidate list; 2) consensus information across different engines is useful for improving the ranking result.

Table 3 (from this section to save space we show results in Test-2 set as it contains more testing questions) shows pruning results on single and multiple extraction engines with the same pruning parameters. When combining all the engines' results, recall is increased. Pruning helps to reduce noises, but it also hurts recall. When independently prune each engine's candidate list, their recalls all degrade, and the combination's (labeled as Combine with Single Prune) recall also degrades seriously. As for our method (labeled as Combine with Joint Prune), when we consider the global appearances from different engines, the impact from pruning on answer recall is eased.

Table 4 shows ranking results of single engine. Single engine's ranker is trained on their own candidate list with the same model and features except consensus features. After combining results from different engines, accuracy of answer extraction is improved comparing to any single one. But if we remove consensus features during combination, performance of answer ranking drops a lot. Further, if we remove pruning based on redundancy among different engines, the

⁴ We didn't compare our method with Watson's extraction method, as Watson employs lots of manual efforts and hundreds of features, which makes directly comparison impossible. It is also why previous work [2,3] don't make such comparison.

Table 3. Results of pruning on single and multiple extraction engines on Test-2. Recall is the binary recall of correct answer, N:P indicates ratio of negative cases' number to positive ones'.

	NP		Wiki		CRF		Combine (Single Prune)		Combine (Joint Prune)	
	Recall	N:P	Recall	N:P	Recall	N:P	Recall	N:P	Recall	N:P
No Pruning	93.52	23	85.32	70	86.69	15	94.54	59	94.54	59
Pruning	82.26	15	77.37	37	79.18	9	85.67	29	93.86	31

Table 4. Performance of ranking results on Test-2 set

Method	Top 1 Acc.	Top 5 Acc.	MRR
NP	53.24	73.72	62.04
Wiki	51.88	71.67	60.36
CRF	54.27	76.45	63.67
All	66.55	79.52	69.93
All -Consensus Features	56.31	75.43	63.97
All -Joint Pruning	54.61	77.82	64.39

result also becomes worse. In all, employing multiple extraction engines helps to increase the performance of answer extraction for Web-based QA.

Table 5 shows an example in testing set with top 5 candidates. Tree kernel and NP method tend to extract complete units from search texts, so they share some common candidates; CRF method tends to extract and give high rank to short candidates related to question, such as years in this example; Wikititle-based method extracts entities in snippets, such as years, titles in this case. They perform different analysis on the texts thus the results present with different trends. Although none of the engine correctly answer the question, after combining the three lists, our method outputs the correct answer.

Feature Ablation Test

We compare contributions from different feature sets by removing one feature set at a time and performing same training and testing. Results are shown in Table 6, removing features in ranking does not have impact on recall, so we only show accuracy measurements. Most of the features are similarity features, so that set has the most contribution. We only design shallow similarity features, if we employ more syntactic features, ranking performance can be further improved. Besides, consensus as illustrated before also has very important role. Follow is redundancy features, from the results we see that redundancy with only 7 features can greatly improve ranking results. Last, quality and search features, although with very few number of features, are also useful.

Table 5. Example of different extraction engines' results

Question	When did Jack Welch retire from GE?
Answer	2001
Method	Top 5 Candidates
Tree kernel	general electric; chairman and ceo; his younger wife; oct 05, 2012;2001
NP	general electric; jack welch 's; 2001; one; chairman and ceo
CRF	2012; 2002; 2001; one; 1999
Wiki	general electric; retirement; 2012; 2001; ceo
Ours	2001;general electric; 2012; retirement; ceo

Table 6. Testing results of ablating different feature sets on Test-2 set

Feature Set	Number of Features	Top 1 Acc.	Top 5 Acc.	MRR
All	73	66.55	79.52	69.93
-Similarity	73-41=32	54.27	74.06	62.05
-Consensus	73-9=64	56.31	75.43	63.97
-Redundancy	73-7=66	57.34	76.79	64.74
-Quality	73-9=64	63.83	77.47	68.46
-Search	73-7=66	64.51	77.82	68.51

Error Analysis

We randomly select 50 wrongly answered questions and analyze the errors. 5 questions' answers are missed in the hypothesis space, most of them are Quality or Time questions. 15 questions are correctly answered, but the labeled answer is different from our output. For example, for question *Where did the Battle of the Bulge take place?*, the given answer is *Luxembourg*, but our output *Ardennes* is also correct. In traditional QA, answers are set based on given corpus, so the expression of the correct answer is fixed; but in Web, the way to express the answer are various, strictly judge answer's correctness based on TREC data will underestimate performance of Web-based QA. Further, 30 questions' answers are wrongly ranked which suggests using more sophisticated ranking features.

5 Conclusion

Web-based Question Answering is to generate answer from search snippets returned by search engines. Search snippets contain many noises and many incomplete sentences, which lowers down performance of traditional methods of answer extraction. In this paper we discuss about using multiple extraction engines for Web-based QA. We adopt a Pruned Rank Aggregation method to prune noisy candidates with redundancies among different engines during rank aggregation. The resulted candidate list is ranked with a Learning to Rank method with similarity, redundancy, search and quality features. Our method improves performance of Web-based answer extraction, and outperforms state-of-the-art answer extraction method.

Acknowledgments. We'd like to thank Yajuan Duan for her contribution of question analysis components. We also want to thank all the reviewers for their valuable comments.

References

1. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. In: TREC, pp. 393–400 (2001)
2. Yao, X., Van Durme, B., Callison-Burch, C., Clark, P.: Answer extraction as sequence tagging with tree edit distance. In: HLT-NAACL, pp. 858–867 (2013)
3. Severyn, A., Moschitti, A.: Automatic feature engineering for answer selection and extraction. In: EMNLP, pp. 458–467 (2013)
4. Sun, H., Duan, N., Duan, Y., Zhou, M.: Answer extraction from passage graph for question answering. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2169–2175. AAAI Press (2013)
5. Xu, J., Licuanan, A., May, J., Miller, S., Weischedel, R.: Answer selection and confidence estimation. In: 2003 AAAI Symposium on New Directions in QA (2003)
6. Ravichandran, D., Ittycheriah, A., Roukos, S.: Automatic derivation of surface text patterns for a maximum entropy based question answering system. In: Proceedings of HLT-NAACL (2003)
7. Sasaki, Y.: Question answering as question-biased term extraction: A new approach toward multilingual qa. In: Proceedings of ACL, pp. 215–222 (2005)
8. Bunescu, R., Huang, Y.: Towards a general model of answer typing: Question focus identification. In: Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics, RCS Volume, pp. 231–242 (2010)
9. Chu-Carroll, J., Fan, J.: Leveraging wikipedia characteristics for search and candidate generation in question answering. In: Proceedings of AAAI (2011)
10. Lin, J.: An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems* 25(2), 6 (2007)
11. Subbian, K., Melville, P.: Supervised rank aggregation for predicting influence in networks. arXiv preprint arXiv:1108.4801 (2011)
12. Agarwal, A., Raghavan, H., Subbian, K., Melville, P., Lawrence, R.D., Gondek, D.C., Fan, J.: Learning to rank for robust question answering. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 833–842. ACM (2012)
13. Joachims, T.: Training linear svms in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 217–226. ACM (2006)
14. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., et al.: Building watson: An overview of the deepqa project. *AI Magazine* 31(3), 59–79 (2010)
15. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, 2493–2537 (2011)
16. Shi, S., Liu, X., Wen, J.R.: Pattern-based semantic class discovery with multi-membership support. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1453–1454. ACM (2008)
17. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)

Answering Natural Language Questions via Phrasal Semantic Parsing*

Kun Xu, Sheng Zhang, Yansong Feng**, and Dongyan Zhao

Peking University, Beijing, China

{xukun, evancheung, fengyansong, zhaodongyan}@pku.edu.cn

Abstract. Understanding natural language questions and converting them into structured queries have been considered as a crucial way to help users access large scale structured knowledge bases. However, the task usually involves two main challenges: recognizing users' query intention and mapping the involved semantic items against a given knowledge base (KB). In this paper, we propose an efficient pipeline framework to model a user's query intention as a phrase level dependency DAG which is then instantiated regarding a specific KB to construct the final structured query. Our model benefits from the efficiency of linear structured prediction models and the separation of KB-independent and KB-related modelings. We evaluate our model on two datasets, and the experimental results showed that our method outperforms the state-of-the-art methods on the Free917 dataset, and, with limited training data from Free917, our model can smoothly adapt to new challenging dataset, WebQuestion, without extra training efforts while maintaining promising performances.

1 Introduction

As very large structured knowledge bases have become available, e.g., YAGO [2], DBpedia [3] and Freebase[4], answering natural language questions over structured knowledge facts has attracted increasing research efforts. Different from keyword based information retrieval, the structure of query intentions embedded in a user's question can be represented by a set of predicate-argument structures, e.g., $\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle$ triples, and effectively retrieved by a database search engine. Generally, the main challenge of understanding the query intention in a structural form is to solve two tasks: recognizing the predicate-argument structures and then instantiating these structures regarding a given KB.

Considering the example question shown in Figure 1, the structure of the query intention consists of multiple predicate-argument pairs, involving an named entity *france* mapping to a KB entity "France", a word *country* mapping to a KB type "Country" and a verb *colonise* possibly indicating a KB relation a *country* "*/colonise*" another *country*. Intuitively, the two subtasks would be solved in a joint framework, e.g., [14]

* This work was supported by the National High Technology R&D Program of China (Grant No. 2012AA011101, 2014AA015102), National Natural Science Foundation of China (Grant No. 61272344, 61202233, 61370055) and the joint project with IBM Research.

** Corresponding author.

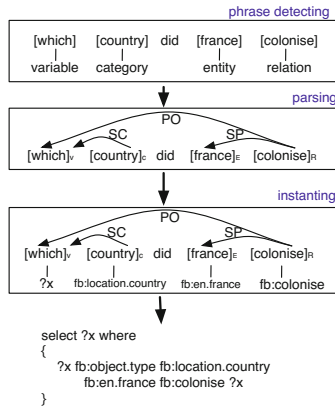


Fig. 1. An example of converting a natural language question into a structured query via phrasal semantic parsing

proposed a PCFG-based semantic parser to simultaneously learn the combination rules among words or phrases and the mappings to specific KB components. However, given the size of existing KBs (usually thousands of predicates, millions of entities and billions of knowledge facts), it makes difficult to jointly train such a PCFG-based parser (the model of [14] takes several days to train with 3,000 sentences), and even more difficult to adapt to other KBs, let alone retrieving multiple KBs within one query, e.g., some queries in the QALD task[6] are mixed with predicates from both DBpedia and Yago. In contrast, we find that recognizing the query intention structure is usually KB-independent. Take Figure 1 as an example, without grounding to a knowledge base, we can still guess that a location called *france* has some relationship, indicated by the verb “colonise”, with some *countries*, (the queried objects), which can be learned directly without reliance on a specified KB. On the other hand, the task of mapping semantic phrases from the intention structures to items in a given KB and producing the final structured queries is KB-dependent, since one has to solve these mappings according to the schema of a specified KB.

Given the observations above, we thus assume that the structure of a question’s query intention can be learned independent from a specific knowledge base, while grounding and converting a query intention into a structured query is dependent on a knowledge base. Our assumption will naturally lead to a pipeline paradigm to translating a natural language question into a structured query, which can then be directly retrieved by a structured database query engine, e.g., Virtuoso¹.

In this paper, we deal with the task of understanding natural language questions in a pipeline paradigm, involving mainly two steps: recognizing the query intention structure inherent in the natural language questions, and then instantiating the query intention structures by mapping the involved semantic items into existing KBs. In the first phase, we build a phrase detector to detect possible semantic phrases, e.g., variables, entity phrases, category phrases and relation phrases. We then develop a semantic parser to

¹ <http://www.virtuoso.com>

predict the predicate-argument structures among phrases to represent the structure of query intentions. In the second phase, given the intention structures, we are then able to adopt a structured perceptron model to jointly solve the mappings between semantic phrases and KB items. By taking a two-phase format, our proposed model can benefit from the separation of KB related components and KB independent steps, and recognize the intention structures more efficiently while making the KB-related component flexible, e.g., we can only retrain the second phase when adapting to new KBs, which is similar in spirit with [13], who rely on a CCG parser to produce an ontological-independent logical representation to express users' intention. We evaluate our model on three datasets, and show that our model can effectively learn the structures of query intentions, and outperform the state-of-the-art methods in terms of question-answering accuracy. Specifically, by testing on a new dataset with large and broad-coverage KB predicates, our model can still perform comparably to the state of the arts without any extra training on the new datasets. By just adjusting the KB related components, our model can maintain a promising results on a new KB.

This rest of the paper is organized as follows. We first briefly describe related work in Section 2. Our Task definition is introduced in Section 3. Section 4 and 5 describe the two steps of our framework: recognizing the structure of query intention and instantiating query intention regarding KB. Our experiments are presented and discussed in Section 6. We finally conclude this paper in Section 7.

2 Related Work

Question answering is a long-standing problem in the field of natural language processing and artificial intelligence. Previous research is mainly dominated by keyword matching based approaches, while recent advancements in the development of structured KBs and structured query engines have demanded the research of translating natural language questions into structured queries, which can then be retrieved using a structured query engine. Existing methods can be roughly categorized into two streams, pattern/template-based models [8–10] and semantic parsing-based models [11–15].

[8] use lexical-conceptual templates for query generation but do not address the disambiguation of constituents in the question. [16] rely on a manually created ontology-driven grammar to directly map questions onto the underlying ontology, where the grammars are hard to adapt or generalize to other large scale knowledge bases. They further develop a template-based approach to map natural language questions into structured queries[10]. [9] collect the mapping between natural language expressions and Yago2 predicates using a set of predefined patterns over dependency parses, and find an optimal mapping assignments for all possible fragments in the questions using an ILP model. Those methods are mainly reply on a set of manually created templates or patterns to collect lexicons or represent the structure of query intentions, therefore are difficult to scale in practice due to the manual efforts involved.

[11] use distant supervision to collect training sentences as well as manual rules to construct CCG lexicons from dependency parses in order to train a semantic parser. [12] develop a probabilistic CCG-based semantic parser, FreeParser, where questions are automatically mapped to logical forms grounded in the symbols of certain fixed ontology

or relational database. They take a similar distant supervision approach to automatically construct CCG lexicon and induce combination rules [17], though with inadequate coverage, for example, their parser will fail if any phrase in the question is not included in the lexicon of the PCCG parser. [14] develop a PCFG-based semantic parser, where a *bridge* operation is proposed to improve coverage and they utilize a set of manual combination rules as well as feature-simulated *soft rules* to combine predicates and produce logical forms.

To handle the *mismatch* between language and the KB, [13] develop a PCCG parser to build an ontology-independent logical representation, and employ an ontology matching model to adapt the output logical forms for each target ontology. [15] first generate candidate canonical utterances for logical forms, then utilize paraphrase models to choose the canonical utterance that best paraphrases the candidate utterance, and thereby the logical form that generated it.

In contrast, we focus on translating natural language questions into structured queries by separating the KB independent components from the KB-related mapping phase. Like [12], our model takes question-phrase dependency DAG pairs as input for our structure recognition phase, but relies far less training data than [12] towards a open domain parser, since we do not learn KB related mappings during structured predictions. We then learn a joint mapping model to instantiate the phrase dependency DAG with a given KB. Our model is simple in structure but efficient in terms of training, since we have a much smaller search space during structure prediction with respect to the query intention, and still hold the promise for further improvement, for example, taking question-answer pairs as training data after initializing with some question-DAG training samples.

3 The Task

We define the task of using a KB to answer natural language questions as follows: given a natural language question q_{NL} and a knowledge base KB, our goal is to translate q_{NL} into a structured query in certain structured query language, e.g., SPARQL, which consists of multiple triples: a conjunction of <subject, predicate, object> search conditions.

4 Recognizing the Structure of Query Intention

Our framework first employs a pipeline of phrase detection and phrase dependency parsing to recognize the inherent structure of user's query intention, which is then instantiated regarding a specific KB.

4.1 Phrase Detection

We first detect phrases of interest that potentially correspond to semantic items, where a detected phrase is assigned with a label $l \in \{entity, relation, category, variable\}$. Entity phrases may correspond to entities of KB, relation phrases correspond to KB's

predicates and category phrases correspond to KB’s categories. This problem can be casted as a sequence labeling problem, where our goal is to build a tagger to predict labels for a sentence. For example:

what are the sub-types of coal
V-B none R-B R-I R-I E-B

(Here, we use *B-I* scheme for each phrase label: *R-B* represents the beginning of a relation phrase, *R-I* represents the continuation of a relation phrase). We use structured perceptron[18] to build our phrase tagger. Structured perceptron is an extension to the standard linear perceptron for structured prediction. Given a question instance $x \in X$, which in our case is a sentence, the structured perceptron involves the following *decoding problem* which finds the best configuration $z \in Y$, which in our case is a label sequence, according to the current model w :

$$z = \arg \max_{y' \in Y(x)} w \cdot f(x, y')$$

where $f(x, y')$ represents the feature vector for instance x along with configuration y' . We use three types of features: lexical features, POS tag features and NER features. Table 1 summarizes the feature templates we used in the phrase detection.

Table 1. Set of feature templates for phrase detection

p = pos tag; n = ner tag; w = word; t = phrase type tag; i = current index

1	unigram of POS tag	p_i
2	bigram of POS tag	$p_i p_{i+1}, p_{i-1} p_i$
3	trigram of POS tag	$p_i p_{i+1} p_{i+2}, p_{i-1} p_i p_{i+1}, p_{i-2} p_{i-1} p_i$
4	unigram of NER tag	n_i
5	bigram of NER tag	$n_i n_{i+1}, n_{i-1} n_i$
6	trigram of NER tag	$n_i n_{i+1} n_{i+2}, n_{i-1} n_i n_{i+1}, n_{i-2} n_{i-1} n_i$
7	unigram of word	w_i
8	bigram of word	$w_i w_{i+1}, w_{i-1} w_i$
9	trigram of word	$w_i w_{i+1} w_{i+2}, w_{i-1} w_i w_{i+1}, w_{i-2} w_{i-1} w_i$
10	previous phrase type	t_{i-1}
11	conjunction of previous phrase type and current word	$t_{i-1} w_i$

4.2 Phrase Dependency Parsing with Multiple Heads

As shown in Figure 1, query intention can be represented by dependencies between “country”, “france” and “colonise”, forming a phrase dependency DAG, we thus introduce a transition-based DAG parsing algorithm to perform a structural prediction process and reveal the inherent structures.

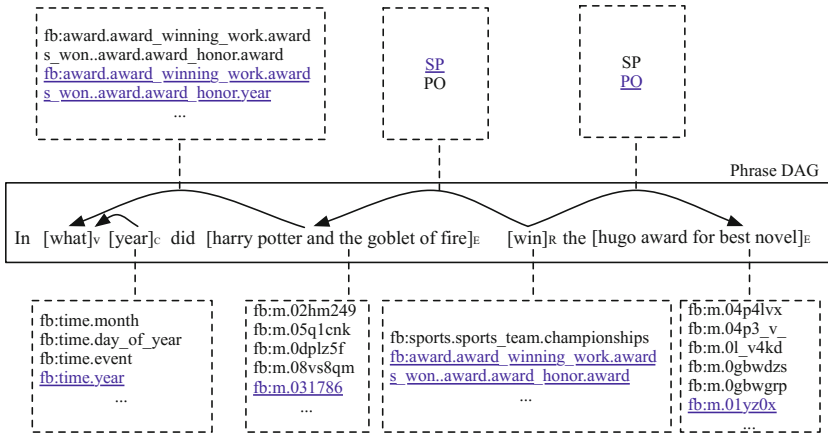


Fig. 2. An example of phrasal semantic DAG, where the dashed boxes list the mapping candidates for all phrases and the underlined are the gold-standard mappings.)

Phrase Dependency DAG. We propose to use the predicate-argument dependencies to capture the query intention, that is, the arguments of a predicate are dependents of that predicate. Here, each predicate is either a unary predicate (characterize its only argument) or a binary predicate (represents the semantic relation between its two arguments). For example, in Figure 2, the category phrase “year” indicates the variable is one specific year, and the relation phrase “win” indicates that the award “hugo award for best novel” is won by “harry potter and the goblet of fire”.

Phrase Dependency Parsing. Note that, in our setup, one phrase can have more than one head, as in Figure 2, variable node *what* has two heads in the resulting dependency DAG. We thus use the framework proposed by [19], i.e., extending traditional arc-eager shift-reduce parsing with multiple heads to find a DAG directly. Specifically, given a question with sequence of phrases, our parser uses a stack of partial DAGs, a queue of incoming phrases, and a series of actions to build a dependency DAG. We assume that each input phrase has been assigned a POS-tag and a semantic label.

Our semantic parser uses four actions: SHIFT, REDUCE, ARCRIGHT and ARCLEFT.

The SHIFT action follow the standard definitions that just pushes the next incoming phrase onto the stack.

The REDUCE action pops the stack top. Note that, the standard REDUCE action which is taken on the condition that the stack top has at least one head. This precondition ensures the dependency graph is a connected graph. However, our phrase dependency parser only concerns the predicate-argument structures, and we add a dependency only between the predicate and argument of our interest. In our case, the dependency graph can be a unconnected directed graph.

The ARCRIGHT action adds a dependency edge from the stack top to the first phrase of the incoming queue, where the phrase on the stack is the head and the phrase in the

Algorithm 1. The decoding algorithm for the phrase DAG parsing; K is the beam size

Require: sentence x

agenda: hold the K -best candidate items

Ensure: *candidate_output*

```

1: agenda.clear()
2: agenda.insert(GetStartItem( $x$ ))
3: candidate_output = NONE
4: while not agenda.empty() do
5:   list.clear()
6:   for all  $item \in agenda$  do
7:     for all  $action \in getActions(actions, item)$  do
8:        $item' = item.apply(action)$ 
9:       if  $item'.F == TRUE$  then
10:        if candidate_output == NONE
            or  $item'.score > candidate\_output.score$  then
11:          candidate_output =  $item'$ 
12:        end if
13:      else
14:        list.append( $item'$ )
15:      end if
16:    end for
17:  end for
18:  agenda.clear()
19:  agenda.insert(list.best( $K$ ))
20: end while

```

queue is the dependent (the stack and queue are left untouched), as long as a left arc does not already exist between these two phrases.

The ARCLEFT action adds a dependency edge from the first phrase on the queue to the stack top, where the phrase in the queue is the head and the phrase on the stack is the dependent (again, the stack and queue are left untouched), as long as a right arc does not already exist between the two phrases.

The Decoding Algorithm for Phrase DAG Parsing. We apply the standard beam-search along with early-update to perform inexact decoding [20] during training. To formulate the decoding algorithm, we define a *candidate item* as a tuple $\langle S, Q, F \rangle$, where S represents the stack with partial derivations that have been built, Q represents the queue of incoming phrases that have not been processed, and F is a boolean value that represents whether the candidate item has been finished. A candidate item is finished if and only if the queue is empty, and no more actions can be applied to a candidate item after it reaches the finished status. Given an input sentence x , we define the start item as the unfinished item with an empty stack and the whole input sentence as the incoming phrases (line 2). A derivation is built from the start item by repeated applications of actions (SHIFT, REDUCE, ARCLEFT and ARCRIGHT) until the item is finished.

To apply beam-search, an agenda is used to hold the K -best partial (unfinished) candidate items at each parsing step. A separate *candidate output* is used to record the

current best finished item that has been found, since candidate items can be finished at different steps. Initially the agenda contains only the start item, and the candidate output is set to none(line 3). At each step during parsing, each candidate item from the agenda is extended in all possible ways by applying one action according to the current status(line 7), and a number of new candidate items are generated(line 8). If a newly generated candidate is finished, it is compared with the current *candidate output*. If the candidate output is none or the score of the newly generated candidate is higher than the score of the *candidate output*, the *candidate output* is replaced with the newly generated item(line 11); otherwise the newly generated item is discarded (line 14). If the newly generated candidate is unfinished, it is appended to a list of newly generated partial candidates. After all candidate items from the agenda have been processed, the agenda is cleared(line 18) and the K-best items from the list are put on the agenda(line 19). Then the list is cleared and the parser moves on to the next step. This process repeats until the agenda is empty (which means that no new items have been generated in the previous step), and the candidate output is the final derivation. Pseudocode for the decoding algorithm is shown in Algorithm 1.

Table 2. The set of feature templates used in our phrase DAG parser

p = phrase; t = POS-tag; s = phrase type

Category	Description	templates
lexical features	stack top	$STpt; STp; STt;$
	current phrase	$N_0pt; N_0p; N_0t$
	next phrase	$N_1pt; N_1p; N_1t;$
	ST and N0	$STptN_0pt; STptN_0p;$
	POS bigram	N_0tN_1t
	POS trigrams	$N_0N_1tN_2t;$
	N0 phrase	$N_0pN_1tN_2t;$
semantic features	Conjunction of phrase label and pos tag	$N_0s; N_0ts; N_0ps;$ $N_1s; N_1ts; STtN_0s;$ $STsN_0t; STpN_0s;$ $STtN_0t; STsN_0s;$
structural features	Indicates whether exists an arc between the stack top item and next input item, and if so what type of arc	$ArcLeft(STs, N_0s);$ $ArcRight(STs, N_0s)$

Features. Features play an important role in transition-based parsing. Our parser takes three types of features: lexical, semantic and structure-related features. We summarize our feature templates in Table 2, where ST represents the top node in the stack, N_0 , N_1 , N_2 represent the three incoming phrases from the incoming queue, subscript t indicates POS tags, subscript p indicates lexical surface forms and subscript s represent the semantic label of the phrase (*entity, relation, category and variable*).

Lexical features include features used in traditional word level dependency parsing with some modifications: all co-occurrences are built on phrase nodes and the POS tag of a phrase is defined as the concatenation of each token’s POS tag in the phrase.

Note that ARCLEFT and ARCRIGHT actions are considered only when the top phrase of the stack and the next phrase are variable, entity or relation phrases. To guide the ARCLEFT and ARCRIGHT actions, we introduce semantic features indicating the semantic label of a phrase.

Recall that, our phrase semantic DAG parser allows one phrase to have multiple heads. Therefore, we modify the ARCLEFT and ARCRIGHT actions so that they can create new dependency arcs without removing the dependent from further consideration for being a dependent of other heads. We thus introduce new structure-related features to indicate whether an arc already exists between the top phrase on the stack and the next phrase on the queue.

5 Instantiating Query Intention Regarding Existing KBs

Given the query intention represented in the phrase dependency DAG, we need to convert it into a structured query Q_{ind} , which can be then grounded to a *KB*-related database query Q_d via mapping the natural language phrases to the semantic items in the *KB*, e.g., Freebase. However, each phrase in Q_{ind} can be potentially mapped to multiple candidate items of Freebase, as shown in Figure 2. Given a knowledge base *KB* and the Q_{ind} that consists of n triples, we will have:

$$Q_d^* = \arg \max P(Q_d|Q_{ind})$$

For the simplicity of computation, we made necessary independent assumptions, and approximate $P(Q_d|Q_{ind})$ as:

$$\overline{P}(Q_d|Q_{ind}) = \prod_{i=1}^n \overline{P}(s_{d_i}|s_{ind_i})\overline{P}(o_{d_i}|o_{ind_i})\overline{P}(p_{d_i}|p_{ind_i})$$

where the (s, p, o) corresponds to the three parts of a query triple: the subject s , predicate p and object o . In practice, we use the Freebase search API² to compute the probabilities of mapping the subject and object phrase. All subject and object phrases are sent to this API, which returns a ranked list of relevant items with their scores. We normalize the score by the sum of all candidate scores.

Inspired by ?, we apply the Naive Bayes model to compute the probability of mapping the relation phrase:

$$\begin{aligned} \overline{P}(p_d|p_{ind}) &= \overline{P}(p_{ind}|p_d)\overline{P}(p_d) \\ &= \prod_w \overline{P}(w|p_d)\overline{P}(p_d) \end{aligned}$$

² <https://developers.google.com/freebase/>

where the w is the word in the phrase p_{ind} . We used the dataset contributed by [21] additionally with type constraints to estimate $\overline{P}(p_d)$ and $\overline{P}(w|p_d)$. For instance, given the subject `/en/france` that has a Freebase type `/location/country` and the object `/en/French` that has a Freebase type `/language/human.language`, the target p_d should take a subject of type `/location/country` and an object of type `/language/human.language`, which are the type constraints in this case. In this case, the candidates of p_d only contains two properties: `/location/country/official_language` and `/location/country/languages_spoken`.

6 Experiments

6.1 Experimental Setup

The Free917 dataset [12] contains 917 questions annotated with logical forms grounded to Freebase. Note that in all of our experiments, we only use the training set of Free917 as our training data. To prepare the training data for our parser, we first parse these questions with CCG parser and accordingly replace the KB items in the gold-standard logical forms with natural language phrases, which we manually assign a semantic label to. Following [12], we held out 30% of the data for the final test, and perform 3 random 80%-20% splits of the training set for development.

The WebQuestions dataset [14] contains 5,810 question-answer pairs, with the same training/testing split with previous work. This dataset was created by crawling questions through the Google Suggest API, and then obtaining answers through Amazon Mechanical Turk. We directly employ the parser trained on the Free917 to test on the test set of WebQuestions and retrieve the answers by executing the queries against a copy of Freebase using the Virtuoso engine.

When evaluating on the Free917 dataset, we use the *parsing accuracy*, *instantiation accuracy* and *system accuracy* as the evaluation metrics. Specifically, the *parsing accuracy* evaluates our parsing phase, which converts the question into a KB-independent logical form. The *instantiation accuracy* evaluates the mappings from a KB-independent logical form to a KB-related database query. The *system accuracy* evaluates the whole pipeline approach, which converts a question into a KB-related database query. Note that we also evaluate gold answers for this dataset, since the queries with aggregation functions may differ from the gold queries. Considering the WebQuestions dataset only contains the gold answers, we use the F-measure as the system accuracy to evaluate our approach.

Table 3. Results on test sets of Free917 and WebQuestions

	Free917	WebQuestions
CY13	59.0%	-
BCFL13	62.0%	35.7%
KCAZ13	68.0%	-
BCFL14	68.5%	39.9%
Our work	69.0%	39.1%

6.2 Main Results

Table 3 represents results on the test set. Our main empirical result on the Free917 dataset is that our system obtains an answer accuracy of 69.0% on the test set, outperforming 59% reported by [12], 62% reported by [14], and 68.5% reported by [15].

Note that, the Free917 dataset covers only 635 Freebase predicates and are only about Freebase. We thus evaluate our parser on a more natural dataset, WebQuestions, introduced by [14]. WebQuestions consists of 5,810 question-answer pairs involved more domains and relations.

We experiment on the original test set, directly using the parser trained on Free917 to predict the query intention regarding Freebase. Interestingly, our system is able to achieve a relative higher accuracy of 39.1% given the fact that the WebQuestions datasets covers a larger and broader set of predicates in Freebase, indicating that the KB-independent structured predictions can be learned separately from the KB-related mappings, while maintaining a comparable performance. In other words, these experiments show that regardless of the topics people are asking, their way of presenting the questions are still similar, which can be captured by learning the structures of phrases of semantic meanings.

6.3 Error Analysis

We analyzed the WebQuestions and the Free917 examples and found several main causes of errors: (i)Phrase detection also accounts for many errors, e.g., the entity phrase detected in “*What is the name of justin bieber brother*” is “*justin biber brother*” and not “*justin biber*”. This detection error is propagated to phrasal semantic parsing (ii)the probabilistic mapping model may map a relation phrase to a wrong predicate, which is mainly due to the limited coverage of our mapping resources. (iii)our system is unable to handle temporal information, which causes errors in questions like “*what kind of government did the united states have after the revolution*” in WebQuestions, where we fail to recognize “*after the revolution*” as a temporal constraint, due to the fact that we never saw this syntax structures in our training data.

7 Conclusion and Future Work

In this paper, we propose a novel framework to translate natural language questions into structural queries, which can be effectively retrieved by structured query engines and return the answers according a KB. The novelty of our framework lies in modeling the task in a KB-independent and KB-related pipeline paradigm, where we use phrasal semantic DAG to represent users’ query intention, and develop a KB-independent shift-reduce DAG parser to capture the structure of the query intentions, which are then grounded to a given KB via joint mappings. This gives the advantages to analyze the questions independent from a KB and easily adapt to new KBs without much human involvement. The experiments on two datasets showed that our model outperforms the state-of-the-art methods in Free917, and performs comparably on a new challenging dataset without any extra training or resources.

Currently, our model requires question-DAG pairs as the training data, though we do not need to annotate new data for every datasets or KBs, it is still promising to extend

our model to train on question-answer pairs only, further saving human involvement. Secondly, in the pipeline of phrase detection and phrase dependency parsing, error propagated from the upstream to downstream. A joint model with multiple perceptrons may help to eliminate the error propagation.

References

1. Fader, A., Soderland, S., Etzioni, O.: Identifying Relations for Open Information Extraction. In: EMNLP, pp. 1535–1545 (2011)
2. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: WWW, pp. 697–706 (2007)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., et al. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
4. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: SIGMOD, pp. 1247–1250 (2008)
5. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic Parsing on Freebase from Question-Answer Pairs. In: EMNLP, pp. 1533–1544 (2013)
6. Cimiano, P., Lopez, V., Unger, C., Cabrio, E., Ngonga Ngomo, A.-C., Walter, S.: Multilingual Question Answering over Linked Data (QALD-3): Lab Overview. In: Forner, P., Müller, H., Paredes, R., Rosso, P., Stein, B. (eds.) CLEF 2013. LNCS, vol. 8138, pp. 321–332. Springer, Heidelberg (2013)
7. Kwiatkowski, T., Choi, E., Artzi, Y., Zettlemoyer, L.S.: Scaling Semantic Parsers with On-the-Fly Ontology Matching. In: EMNLP, pp. 1545–1556 (2013)
8. Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jörg, B., Schäfer, U.: Question answering from structured knowledge sources. *J. Applied Logic*, 20–48 (2007)
9. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Natural Language Questions for the Web of Data. In: EMNLP-CoNLL, pp. 379–390 (2012)
10. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: WWW, pp. 639–648 (2012)
11. Krishnamurthy, J., Mitchell, T.: Weakly Supervised Training of Semantic Parsers. In: EMNLP-CoNLL, pp. 754–765 (2012)
12. Cai, Q., Yates, A.: Semantic Parsing Freebase: Towards Open-domain Semantic Parsing. In: SEM (2013)
13. Kwiatkowski, T., Choi, E., Artzi, Y., Zettlemoyer, L.S.: Scaling Semantic Parsers with On-the-Fly Ontology Matching. In: EMNLP, pp. 1545–1556 (2013)
14. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic Parsing on Freebase from Question-Answer Pairs. In: EMNLP, pp. 1533–1544 (2013)
15. Berant, J., Liang, P.: Semantic Parsing via Paraphrasing. In: ACL (2014)
16. Unger, C., Cimiano, P.: Pythia: Compositional Meaning Construction for Ontology-based Question Answering on the Semantic Web. In: Muñoz, R., Montoyo, A., Métais, E. (eds.) NLDB 2011. LNCS, vol. 6716, pp. 153–160. Springer, Heidelberg (2011)
17. Kwiatkowski, T., Zettlemoyer, L.S., Goldwater, S., Steedman, M.: Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification. In: EMNLP, pp. 1223–1233 (2010)
18. Collins, M.: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: EMNLP (2002)
19. Sagae, K., Tsujii, J.: Shift-Reduce Dependency DAG Parsing. In: COLING, pp. 753–760 (2008)
20. Collins, M., Roark, B.: Incremental Parsing with the Perceptron Algorithm. In: ACL, pp. 111–118 (2004)
21. Yao, X., Van Durme, B.: Information Extraction over Structured Data: Question Answering with Freebase. In: Proceedings of ACL (2014)

A Fast and Effective Method for Clustering Large-Scale Chinese Question Dataset

Xiaodong Zhang and Houfeng Wang

Key Laboratory of Computational Linguistics, Peking University,
Ministry of Education, China
zxddavy@gmail.com, wanghf@pku.edu.cn

Abstract. Question clustering plays an important role in QA systems. Due to data sparseness and lexical gap in questions, there is no sufficient information to guarantee good clustering results. Besides, previous works pay little attention to the complexity of algorithms, resulting in infeasibility on large-scale datasets. In this paper, we propose a novel similarity measure, which employs word relatedness as additional information to help calculating similarity between questions. Based on the similarity measure and k-means algorithm, semantic k-means algorithm and its extended version are proposed. Experimental results show that the proposed methods have comparable performance with state-of-the-art methods and cost less time.

Keywords: Question Clustering, Word Relatedness, Semantic K-means.

1 Introduction

In recent years, short texts, such as snippets, micro-blogs, and questions etc., are prevalent on the Internet. Community Question Answering (CQA) websites have accumulated large archives of question-answer pairs, which promote the development of the question-answer datasets based QA system. Such QA systems retrieve questions from the dataset, which are semantically equivalent or relevant to queried questions, and show corresponding answers to users. To retrieve questions fast in a large-scale dataset, one feasible way is to cluster questions in advance so as to reduce the retrieval range.

There are two major challenges for question clustering. Firstly, question clustering faces data sparseness problem. Unlike normal texts with lots of words, questions only consist of several sentences (even just a few words). They do not provide sufficient statistical information, e.g. word co-occurrence, for effective similarity measure [1]. Secondly, question clustering suffers from lexical gap problem. Different words are used to express the same meaning in human languages. Conventional methods usually ignore the useful information of literally different but related words. In addition to these two challenges, the complexity of clustering algorithm should be concerned about. There are more than 300 million questions in zhidao¹, the most famous Chinese CQA, and the number keeps

¹ <http://zhidao.baidu.com/>

growing. Time-consuming or space-consuming methods cannot be performed on large-scale datasets without costly hardware resources. It is beneficial that the clustering algorithm is both effective and fast.

In order to settle the problems in question clustering, we first present a novel question similarity measure, which uses word relatedness to bridge the lexical gap between questions. The relation of two questions is modeled as a bipartite graph, based on which we define the similarity of two questions. Next we propose an effective and fast question clustering algorithm, referred to as semantic k-means (Sk-means), by introducing the proposed similarity measure into the k-means algorithm. An extended method, referred to as extended semantic k-means (ESk-means), is presented to improve the effectiveness further and gives more choices of the balance of effectiveness and complexity. We classified 16000 Chinese questions manually and built a classified question dataset to compare the performances of our method and some other popular approaches. The experimental results show that our proposed methods are very successful.

2 Related Works

Many methods have been proposed to improve short text clustering. Some researchers employed name entities [2] and phrases [3] extracted from the original text to construct the feature space, which is called surface representation. Ni [4] presented a novel clustering strategy, TermCut, which recursively select a core term and bisect text graph according to the term. Lack of semantic knowledge, these techniques suffer from the lexical gap problem. Another way is to enrich the text representation based on “bag of words” model by generating external features from linguistic and collaborative knowledge bases. Hotho [5] observed that additional features from WordNet can improve clustering results. Somnath [6] proposed a method to enrich short texts representation with additional features from Wikipedia. However, enriching the representation by knowledge usually require structured knowledge bases, e.g. WordNet and Wikipedia etc., which is scarce in Chinese circumstance.

Topic models have been proposed to uncover the latent semantic structure from text corpus and can be used for clustering. Latent Semantic Analysis (LSA) [7], Probabilistic Latent Semantic Analysis (PLSA) [8] and Latent Dirichlet Allocation (LDA) [9] have been proved successful on normal texts. Hong [10] made a comprehensive empirical study of topic modeling in Twitter, and suggested that new topic models for short texts are in demand. Yan [11] proposed a biterm topic model for short texts, which learn the topics by directly modeling the generation of word co-occurrence patterns. Ji [12] presented a Question-Answer Topic Model (QATM) to learn the latent topics aligned across the question-answer pairs to alleviate the lexical gap problem. Guo [13] proposed a Weighted Textual Matrix Factorization (WTMF) method to model missing words appropriately. The major challenge of topic models is that short texts do not provide sufficient word co-occurrence or other statistics to learn hidden variables. The performance of topic model in short text is not as good as in normal text.

3 Question Similarity Measure

Similarity or distance measure plays an important role in clustering algorithm based on text similarity. Compared to normal text, questions suffer from data sparseness and lexical gap. There are probably only a few (even none) words in common between two related questions. Vector space model (VSM) is the most commonly used text representation method. Based on VSM, it is inaccurate to calculate question similarity by conventional similarity measures, e.g. cosine similarity. Consider the two following Chinese questions:

Question 1: 电脑出故障, 过了保修期怎么办? (My computer broke down and its warranty expired. What should I do?)

Question 2: 我想给笔记本装个固态硬盘, 哪个牌子比较好? (I would like to install a SSD to my laptop. Which brand is good?)

Both about computers, the two questions are highly correlative. However, there are no words literally same between the two questions so that an extremely low correlation is given by cosine similarity. Cosine similarity considers that the relation between words is binary, literally same or not, ignoring different relatedness between words.

We introduce word relatedness as semantic information into our question similarity measure. Many methods were proposed for calculating word relatedness [14, 15]. We use word2vec² to calculate word relatedness in this work. This tool provides an efficient implementation of the continuous bag-of-words and skip-gram architectures [16, 17] for computing vector representations of words. Word relatedness can be defined as cosine similarity of the vector representations, as is shown in (1), where t_1 and t_2 are two words and v_1 and v_2 are their vector representations, respectively. If the training dataset is large enough, most words that do not appear in training dataset are noises and only a small part is neologisms, which have little influence and can be solved by updating training dataset. The major advantage of this method is that the training data only needs unstructured plain texts, which is easier to acquire than structured resources, e.g. semantic dictionary.

$$r(t_1, t_2) = \begin{cases} 1 & t_1 = t_2 \\ 0 & t_1 \text{ or } t_2 \text{ not in training data} \\ \frac{\langle v_1 \cdot v_2 \rangle}{\|v_1\| \|v_2\|} & \text{otherwise} \end{cases} \quad (1)$$

In our method, questions are represented by VSM. There are various choices for term weights in vectors. Here we use the popular TF-IDF weight and the vectors are normalized. Then we model the similarity of two questions by a bipartite graph. Let $G = (U, V, E)$ denote a bipartite graph whose partition has parts U and V , with E denoting the edges of the graph. Let q_1 and q_2 be two questions. The graph is constructed as follows. For each word type t_{1i} ($i \in [1, n_1]$), n_1 is the number of word types in q_1 in q_1 , there is a corresponding node u_i in U . Node v_j in V for each word type t_{2j} in q_2 is defined in the same way. If the

² <https://code.google.com/p/word2vec/>

relatedness of t_{1i} and t_{2j} exceeds a threshold (the detail of threshold setting is discussed in Sect. 5), the two words are considered related and nodes u_i and v_j are connected by a edge, of which the weight is calculated as follows:

$$e_{ij} = w_{1i} \times w_{2j} \times r_{ij} \tag{2}$$

where e_{ij} is the weight of the edge connecting u_i and v_j , w_{1i} and w_{2j} are the TF-IDF weights of word t_{1i} and t_{2j} in vectors, r_{ij} is the relatedness of t_{1i} and t_{2j} . The similarity of two questions is defined as the sum of weights of edges in the maximum weight matching of the bipartite graph. Formally,

$$s = \sum_{e \in \text{MWM}(G)} w_e \tag{3}$$

where $\text{MWM}(G)$ are edges belong to the maximum weight matching of the bipartite graph G and w_e is the weight of edge e .

The maximum weight matching can be solved by Kuhn-Munkres algorithm, with $O(n^3)$ time complexity, where n is the number of nodes in the graph. Although the time complexity is high, the maximum weight matching can be calculated within a short time because n is small in the case of short text (the average number of words in a question is 12.5 in our dataset).

Next we give a concrete example of calculating the similarity of the two questions mentioned above. The vector representations of the two questions are as follows and the decimal after slash is word weight, which is calculated in a true dataset. Question 1: [电脑/0.44, 出/0.11, 故障/0.55, 过/0.17, 保修期/0.67, 怎么办/0.11]; Question 2: [笔记本/0.47, 装/0.27, 固态硬盘/0.80, 哪个/0.07, 牌子/0.33 好/0.07]. Pairs of words that relatedness exceeds 0.2 (threshold) are as follows: (电脑, 笔记本) = 0.78; (电脑, 固态硬盘) = 0.52; (电脑, 装) = 0.21; (故障, 笔记本) = 0.28; (故障, 固态硬盘) = 0.22; (保修期, 笔记本) = 0.28. Fig. 1 shows the bipartite graph of the two questions. The weights of edges (attached to lines) are computed by (2). The maximum weight matching is marked by bold lines and the similarity of the two questions is 0.26, rather than 0 computed by conventional similarity measures, e.g. cosine similarity.

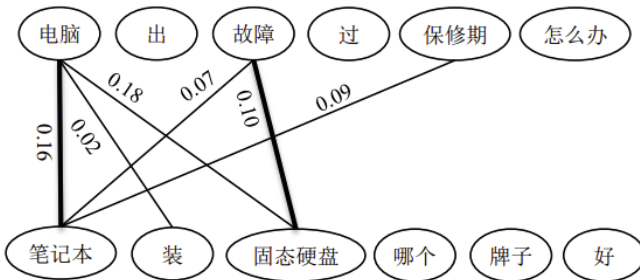


Fig. 1. Bipartite graph of two questions

Our method overcomes the two problems of calculating similarity of short texts. Word relatedness bridges the lexical gap between questions. Literally different but related words can contribute to the similarity calculation. With the help of additional word relatedness information, the sparse features of questions are enriched. Note that we treat questions as general short texts, so our proposed similarity measure can be used in other types of short texts, not just questions.

4 Question Clustering Method

4.1 Semantic K-means Algorithm

Based on our proposed question similarity measure and k-means algorithm, we present a question clustering method, referred to as semantic k-means (Sk-means). The method can also be used for clustering other types of short texts. First of all, we analyze the pros and cons of k-means algorithm in the case of question clustering so that we can present our method naturally.

The major advantage of k-means is low time complexity and space complexity, which is extremely useful for large-scale datasets. However, k-means has poor performance on short texts because it cannot solve the data sparseness and lexical gap problem in questions. In k-means algorithm, k data points are randomly chosen as the initial centroids. In the iterations, every data point is computed the distance to each centroid and then assigned to the closed centroid. Note that in the first iteration the centroid vectors are extremely sparse, which means a data point is hard to decide which centroid is closest to it by conventional similarity measures. A large number of wrong assignments will affect following iterations and eventual results. In the following iterations, however, the centroid vectors become much denser because of re-computing centroids by averaging all data points assigned to the centroids and the similarity calculation is more accurate than in the first iteration. Therefore, the major problem of k-means is at the first iteration. Based on the analysis above, an idea about improving k-means comes up intuitively. Our proposed similarity measure, which employs word relatedness as semantic information to improve question similarity calculation, is used in the first iteration so as to avoid the inaccurate similarity between sparse questions. The cosine similarity is used in the rest iterations to keep the high speed of the algorithm. Algorithm 1 shows the pseudo-code of Sk-means. The questions are represented by VSM so that each question can be regarded as a data point.

The main difference between Sk-means and k-means is that in the first iteration our method employs semantic information (word relatedness) to help similarity calculation of short texts, alleviating the data sparseness and lexical gap. Any stopping criterion used in k-means can be used in our method. In our experiments, the algorithm stops when no data point is assigned to different clusters between two consecutive iterations.

Algorithm 1. semantic k-means (k, D)

```

1: choose  $k$  data points randomly as the initial centroids (cluster centers);
2: repeat
3:   for each data point  $x \in D$  do
4:     if in the first iteration then
5:       compute the similarity of  $x$  and each centroid by our proposed similarity
           measure;
6:     else
7:       compute the similarity of  $x$  and each centroid by cosine similarity
8:     end if
9:     assign  $x$  to the most similar centroid
10:  end for
11:  re-compute the centroid using the current cluster memberships
12: until the stopping criterion is met

```

4.2 Extended Semantic K-means Algorithm

A confusing thing of our algorithm is why the proposed similarity measure is only used in the first iteration. This is mainly because of the balance of effectiveness and efficiency. The proposed similarity measure does help in the following iterations but is not as helpful as in the first iteration. As the centroids become dense in the following iterations, calculating the similarity between data points and centroids by our proposed approach is very time consuming. Therefore, the Sk-means algorithm uses our proposed measure only in the first iteration. Nevertheless, our proposed similarity measure can still help alleviating lexical gap in the following iterations. Thus we give an extended version of Sk-means, referred to as extended semantic k-means (ESk-means), to improve the effectiveness further at cost of higher complexity. The algorithm is shown in Algorithm 2. ESk-means uses our proposed similarity measure in the headmost m iterations, rather than only in the first iteration. A trick is used to reduce time consuming calculations. In the headmost m iterations, we truncate the centroids vectors and only reserve d dimensions with highest weights (all other dimensions are set to 0). Then the truncated vectors are normalized and used for similarity calculation by our proposed measure. In the following iterations, cosine similarity is used and the centroid vectors are not truncated. We can see that ESk-means is just Sk-means if $m = 1$ and the truncation is not performed.

4.3 Complexity Analysis

Next we compare the time and space complexity of our proposed clustering methods with some other clustering methods. The comparisons are shown in Table 1, including six clustering methods, in which BTM [11] is a state-of-the-art topic model for short texts. The notation t is the number of iterations, n is the number of document in dataset, k is the number of clusters, and \bar{l} is average length of a document. In ESk-means, m is the number of iterations using the proposed similarity measure, d is the number of reserved dimensions. In BTM, b

Algorithm 2. extended semantic k-means (k, D, m, d)

```

1: choose  $k$  data points randomly as the initial centroids (cluster centers);
2:  $iter \leftarrow 1$ 
3: repeat
4:   truncate and normalize centroids (reserve  $d$  dimensions)
5:   for each data point  $x \in D$  do
6:     compute the similarity of  $x$  and each centroid by our proposed similarity
       measure;
7:     assign  $x$  to the most similar centroid
8:   end for
9:   re-compute the centroid using the current cluster memberships
10:   $iter \leftarrow iter + 1$ 
11: until  $iter > m$ 
12: repeat
13:   for each data point  $x \in D$  do
14:     compute the similarity of  $x$  and each centroid by cosine similarity;
15:     assign  $x$  to the most similar centroid
16:   end for
17:   re-compute the centroid using the current cluster memberships
18: until the stopping criterion is met

```

is the number of biterns and can be approximately rewritten as $b \approx (n\bar{l}(\bar{l}-1))/2$, and v is the number of word types. Note that different methods have different number of iterations, which is distinguished by subscripts. Usually t_4 and t_5 are much larger than t_1 , t_2 , and t_3 . K-means and our methods usually take tens of rounds before stop, while LDA and BTM take hundreds and even thousands of rounds before stop. \bar{l} is small for questions and can be considered as constant. From the comparisons, we can see that Sk-means has very close time and space complexity with k-means. The complexity of Sk-means is lower than LDA and BTM and the complexity of ESk-means depends on parameter m and d . The complexity of spectral clustering is too high to be used in large-scale datasets. In Sect. 5, we will give the real time consumption and the effectiveness of each method on a dataset.

Table 1. The complexity comparison of proposed methods and some other methods

Method	Time complexity	Space complexity
k-means	$O(t_1 n k \bar{l})$	$O((k+n)\bar{l})$
Sk-means	$O(n k \bar{l}^3 + t_2 n k \bar{l})$	$O((k+n)\bar{l} + \bar{l}^2)$
ESk-means	$O(m n k d^3 + t_3 n k \bar{l})$	$O(n\bar{l} + k d + d^2)$
LDA	$O(t_4 n k \bar{l})$	$O(n k + v k + n \bar{l})$
BTM	$O(t_5 k b)$	$O(k + v k + b)$
spectral clustering	$O(n^2 k)$	$O(n^2)$

5 Experiments

5.1 Corpus and Evaluation Metrics

As far as we know, there is no available large-scale classified Chinese question dataset. So we build a classified Chinese question dataset by collecting questions from zhidao³. Although the questions are classified by users when posting these questions, there is much misclassification for some reasons. We reviewed manually and get rid of the misclassified questions. The question dataset contains 16000 Chinese questions, which are classified into 8 classes and each class consists of 2000 questions. We segment each Chinese question using ICTCLAS⁴ and remove the stop words by checking a stop word list containing 746 Chinese words. Each resultant Chinese word is used as term for further computation.

As for the training data of word2vec, we use 3 datasets acquired from the Internet, including Chinese Wikipedia⁵, Chinese Gigaword⁶, Sogou news corpus⁷. We remove labels and links from the 3 corpus and then segment the remaining text by ICTCLAS. The combination of the 3 corpus is used as training data for word2vec, which calculates the word relatedness.

The *FScore* measure is one of the commonest evaluation metrics for clustering task. We use *Micro Averaged FScore* [4] (denoted as *MicroFScore*) to test the effectiveness of the proposed methods. Due to space limitations, we do not give the definition in this paper.

5.2 Experiments Settings and Results Analysis

We carry out 8 experiments to compare the performance of the proposed methods with other popular methods. The settings are as follows:

Experiment 1: K-means algorithm is carried out, with cosine similarity as similarity measure.

Experiment 2: We perform spectral clustering, which is usually considered a better algorithm than k-means. Cosine similarity is used as similarity measure.

Experiment 3: This experiment is also spectral clustering. The difference from Experiment 2 is that our proposed similarity measure is used. We refer to this experiment as spectral++.

Experiment 4: We enrich the representation of questions with additional features from Wikipedia by the method proposed in [6]. Then k-means is carried out on the enriched representations. This experiment is referred to as wiki.

Experiment 5: LDA is carried out and the hyper parameters are tuned via grid search. In this experiment, $\alpha = 0.1$ and $\beta = 0.05$.

³ <http://zhidao.baidu.com/>

⁴ <http://ictclas.nlpir.org/>

⁵ <http://download.wikipedia.com/zhwiki/>

⁶ <https://catalog.ldc.upenn.edu/LDC2011T13>

⁷ <http://www.sogou.com/labs/dl/ca.html>

Experiment 6: We perform BTM, which is state-of-the-art topic model for short texts. Parameters are also tuned via grid search. In this experiment, $\alpha = 0.1$ and $\beta = 0.01$.

Experiment 7: Sematic k-means algorithm is performed. The threshold in similarity measure is set to 0.3.

Experiment 8: We carry out extended semantic k-means algorithm. The threshold in similarity measure is set to 0.3. In this experiment, $m = 8$ and $d = 50$.

For each experiment, the number of cluster is set to 8, the actual class number. Because of the stochasticity of the algorithms, we perform each algorithm ten times and take the average score of ten times as final score. The consumed time is also the average time of ten times. The performance comparisons are shown in Table 2. The average score, highest score in 10 times and average consumed time are listed in the table.

Table 2. Comparisons of different methods

#	Method	MicroFScore (average)	MicroFScore (highest)	Time
1	k-means	0.644	0.751	6s
2	spectral	0.554	0.575	919s
3	spectral++	0.671	0.721	1725s
4	wiki	0.678	0.757	207s
5	LDA	0.734	0.798	32s
6	BTM	0.741	0.804	148s
7	Sk-means	0.736	0.821	10s
8	ESk-means	0.740	0.821	88s

From the experimental results, we can see that k-means is the fastest method among these methods but the effectiveness is unsatisfying. It is surprising that the score of spectral clustering is even lower than k-means. This is because it is inaccurate to calculate the similarity of questions by conventional similarity measures, resulting in unreliable similarity matrix and eigenvectors. In the iterations of k-means, however, the centroids become dense so that the similarity calculations become accurate. In Experiment 3, the similarity matrix is calculated by our proposed measure and then spectral clustering is performed. Based on this similarity matrix, the result of spectral clustering is better than Experiment 1 and 2, proving that our proposed similarity measure is effective. In Experiment 4, the f-score is 1.4% higher than k-means, showing that enrich the representation by Wikipedia is also helpful. However, the improvement of Experiment 3 and 4 is limited, only slightly higher than Experiment 1, and cost a lot of time. In Experiment 5 and 6, we find that the results of topic model are much better than k-means. In particular, the BTM achieves the highest

performance among all experiments. We test the performance of our methods in Experiment 7 and 8. In Experiment 7, the Sk-means outperforms k-means, spectral clustering, wiki and LDA and the performance is only slightly lower than BTM. The speed of Sk-means is close to k-means. It is amazing that a method with low time complexity can have such good results. In Experiment 8, ESk-means takes more time to gain better results than Sk-means. The performance of ESk-means is comparable to BTM. Although the average score is slightly lower than BTM, the highest score in ten times is achieved by Sk-means and ESk-means. The performance of our proposed methods depends heavily on the initial centroids. For some good initial centroids, our methods can get extremely good results. Unfortunately, we have not found effective methods for choosing good initial centroids. As Sk-means costs little time, if there is way to evaluate (maybe approximately) results, we can run Sk-means many times to get a good result. From all these experiments, we can conclude that Sk-means is a fast and effective method for clustering questions and ESk-means can improve the results further at cost of more time.

The remaining unsettled thing is the setting of some parameters in our methods. First, we discuss the setting of the threshold in our proposed similarity measure. The threshold is used to determine whether two words are considered related. Here we examine the influence of the threshold via Sk-means. We run the algorithm 11 times on the dataset. The threshold is varied from 0 to 1 increased by 0.1 each time. The initial seeds are same in the 11 times. The results are shown in Fig. 2. When the threshold is set around 0.2 and 0.3, the result is good. When the threshold is lower, the information of unrelated words is introduced into the similarity measure and has negative effects. When the threshold is higher, less word relatedness information can be used so that the f-score decreases. As the threshold draws near 1, the Sk-means just becomes k-means actually. Therefore, 0.2 and 0.3 is the recommend threshold value.

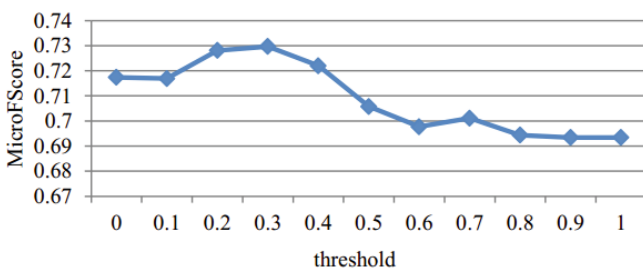


Fig. 2. MicroFscore changes when the threshold is set from 0 to 1

Next we discuss the setting of parameter m and d in ESk-means. An intuitive thought is that larger m and d will produce better results. As shown in Fig. 3, this thought is true within some limits. However, larger m and d mean more consumed time and space. We find that the improvement of f-score is not significant when

m is larger than 8 and d is larger than 50. Therefore, a reasonable choice is setting m to 8 and d to 50. Note that when d is small (e.g. 10), larger m makes the results worse. This is because the centroids consist of too little terms and the truncation makes the centroids lose too much useful information.

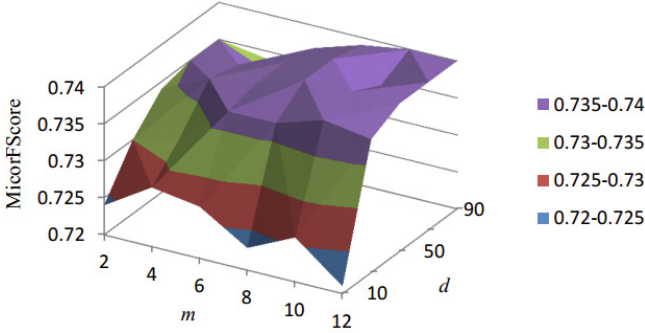


Fig. 3. The influence of parameter m and d in ESk-means

6 Conclusion and Future Work

The CQA websites have accumulated quantities of questions. However, there is no fast and effective clustering method for these large-scale datasets. Our work mainly consists of two parts. Firstly, we propose a novel similarity measure for questions. Word relatedness is employed to tackle the problems of data sparseness and lexical gap in questions. The relation of two questions is modeled by bipartite graph, based on which we define the similarity of two questions. Secondly, we propose Sk-means algorithm and ESk-means algorithm by introducing our proposed similarity measure into k-means algorithm. The experimental results show that Sk-means is a fast and effective method for clustering questions and ESk-means can improve the results further at cost of more time.

There are some interesting future works to be continued. We will explore the application of our similarity measure in other clustering methods and NLP tasks. In consideration of the good results of topic model in the experiments, we will try to introduce word relatedness into topic model to improve question clustering.

Acknowledgments. This research was partly supported by National Natural Science Foundation of China (No 61370117, 61333018) Major National Social Science Fund of China(No.12&ZD227), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101).

References

1. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, pp. 91–100. ACM (April 2008)
2. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 297–304. ACM (July 2004)
3. Chim, H., Deng, X.: Efficient phrase-based document similarity for clustering. *IEEE Transactions on Knowledge and Data Engineering* 20(9), 1217–1229 (2008)
4. Ni, X., Quan, X., Lu, Z., Wenyin, L., Hua, B.: Short text clustering by finding core terms. *Knowledge and Information Systems* 27(3), 345–365 (2011)
5. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 541–544. IEEE (November 2003)
6. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 787–788. ACM (July 2007)
7. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *JASIS* 41(6), 391–407 (1990)
8. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (August 1999)
9. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022 (2003)
10. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics, pp. 80–88. ACM (July 2010)
11. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1445–1456 (May 2013)
12. Ji, Z., Xu, F., Wang, B., He, B.: Question-answer topic model for question retrieval in community question answering. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2471–2474. ACM (October 2012)
13. Guo, W., Diab, M.: Modeling sentences in the latent space. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 864–872. Association for Computational Linguistics (July 2012)
14. Strube, M., Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: *AAAI*, vol. 6, pp. 1419–1424 (July 2006)
15. Gracia, J.L., Mena, E.: Web-based measure of semantic relatedness. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) *WISE 2008*. LNCS, vol. 5175, pp. 136–150. Springer, Heidelberg (2008)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, pp. 3111–3119 (2013)

A Hybrid Method for Chinese Entity Relation Extraction

Hao Wang, Zhenyu Qi*, Hongwei Hao, and Bo Xu

Institute of Automation, Chinese Academy of Sciences, Beijing, China
{h.wang, zhenyu.qi, hongwei.hao, xubo}@ia.ac.cn

Abstract. Entity relation extraction is an important task for information extraction, which refers to extracting the relation between two entities from input text. Previous researches usually converted this problem to a sequence labeling problem and used statistical models such as conditional random field model to solve it. This kind of method needs a large, high-quality training dataset. So it has two main drawbacks: 1) for some target relations, it is not difficult to get training instances, but the quality is poor; 2) for some other relations, it is hardly to get enough training data automatically. In this paper, we propose a hybrid method to overcome the shortcomings. To solve the first drawback, we design an improved candidate sentences selecting method which can find out high-quality training instances, and then use them to train our extracting model. To solve the second drawback, we produce heuristic rules to extract entity relations. In the experiment, the candidate sentences selecting method improves the average F1 value by 78.53% and some detailed suggestions are given. And we submitted 364944 triples with the precision rate of 46.3% for the competition of Sougou Chinese entity relation extraction and rank the 4th place in the platform.

Keywords: Information Extraction, Entity Relation Extraction, Conditional Random Field Model, Knowledge Base.

1 Introduction

Entity relation extraction is one of the main tasks of information extraction. The input of this problem is multi-structured data, including structured data (infobox form), semi-structured data (tables and lists) and non-structured data (free text). And the output is a set of fact triples extracted from input data. For example, given the sentence “姚明出生于上海” (Yao Ming was born in Shanghai) as input, the relation extraction algorithm should extract “<姚明, 出生地, 上海>” (Yao Ming, birthplace, Shanghai) from it. These fact triples can be used to build a large, high-quality knowledge base, which can benefit to a wide range of NLP tasks, such as question answering, ontology learning and summarization.

Now massive Chinese information exists on the internet and the research of Chinese entity relation extraction will have important significance. But current research mainly focuses on the processing of English resource and the study conducted on

* Corresponding author.

Chinese corpus is less. Compared to English language, Chinese language need word segmentation, and the proper nouns don't have the first letter capitalized, so the Chinese entity relation extraction is more difficult and more challenging.

In this paper, we propose a hybrid method for Chinese entity relation extraction. We adopt different methods to extract different frequency relation words. We first build a Chinese semantic knowledge base, using the corpus of Douban web pages, Baidu encyclopedia and Hudong encyclopedia. An improved selecting candidate sentences method trained by conditional random field model is used to extract high-frequency relation words of the knowledge base, and the method based on some simple rules and knowledge base is used to extract low-frequency relation words.

Specifically, our contributions are:

- We propose candidate sentences selecting method, which can reduce the mistakes introduced by automatic tagging training data and improve the extraction performance.
- It's hard to get enough training data for some rare relations. Here, we propose the method based on some simple rules and knowledge base to extract these low-frequency relation words.

The rest of the paper is organized as follows: Section 2 introduces related works of this paper. Section 3 introduces the construction of our Chinese semantic knowledge base, and the improved selecting candidate sentences method trained by conditional random field model for high-frequency relation words, and method based on knowledge base and simple rules for low-frequency relation words. Section 4 describes experimental results and detailed analysis of our methods. We conclude in Section 5.

2 Related Works

Since the late 1980 s, the MUC (Message Understanding Conference) [1] promoted the vigorous development of relation extraction, and made the information extraction to be an important branch in the field of natural language processing. In order to meet the increasing social demand, since 1999, the NIST (National Institute of Standards and Technology) organized ACE (Automatic Content Extraction) reviews [2], and automatically extracting entities, relations, and events in news corpus is the main content of this conference.

Washington University developed TextRunner System [3, 4], the representative of the free text entity relation extraction, and then they released the WOE System [5], which is using Wikipedia for open information extraction. The basic idea of these two systems is first to identify the entity, and then regards the verb between the two entities as relationship. It would have a lot of dislocation, some illogical extraction result, as well as the discontinuous extraction. Continuous analysis and improvement of their previous work, then this group published the second generation open information extraction systems, for example, REVERB [6], R2A2 [7], and OLLIE [8]. Compared to the first generation, the effect of these systems had obvious promotion. The basic idea is first to identify the relationship, and then to identify entities, and specific versions is the result of improvements on the details.

The Intel China research center developed a Chinese named entity extraction system [9], and the relation between these entities can also be extracted. This system obtains rules by memory-based learning algorithm, and then these rules are used to extract named entity and the relationship between them. There have been many previous works of extracting Chinese entity relation by training conditional random field model, and some work use the online encyclopedia corpus [10, 11].

3 Chinese Entity Relation Extraction

Using the structured data part of Baidu encyclopedia and Hudong encyclopedia, we can build a Chinese semantic knowledge base, and this part will be described in detail in section 3.1. Different methods of extracting entity relationship should adapt to the frequency of relation words in our knowledge base.

The improved candidate sentences selecting method trained by conditional random field model, introduced in section 3.2, can be used to extract high-frequency relations. For some low-frequency relations, the method based on simple rules and knowledge base can be used, and we will discuss this case specific to our method for the competition of Sougou Chinese entity relation extraction, and this part will be described in detail in section 3.3. If a relationship is low-frequency in our knowledge base, it is hard to get enough training corpus for the improved conditional random field model, so we need specific treatment for different cases and make full use of the domain knowledge.

3.1 Chinese Semantic Knowledge Base Construction

We gathered the Baidu encyclopedia web data before July 2012 and Hudong encyclopedia web data before October 2012, which are composed of entities, corresponding infobox knowledge and unstructured content. We extract the infobox knowledge from these corpus and represent them in triples format $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$, arg1 and agr2 are represent of entities and rel is represent of relation, like $\langle \text{中国}, \text{首都}, \text{北京} \rangle$, and then store these triples in our knowledge base.

中文名	卧虎藏龙	主演	周润发, 杨紫琼, 章子怡, 张震
外文名	Crouching Tiger, Hidden Dragon	片长	120 min
制片地区	中国, 美国	上映时间	法国: 2000年5月16日
导演	李安	分级	USA:PG-13
编剧	王度庐, 王蕙玲	对白语言	汉语普通话
类型	爱情, 动作, 冒险, 剧情	色彩	彩色
		奖项	奥斯卡最佳外语奖

Fig. 1. An Infobox from Baidu Encyclopedia

Fig 1 shows the infobox knowledge of Baidu encyclopedia, and we can easily extract many triples from the XML files. And the extraction of structured data in Hudong encyclopedia is similar to this.

When we determine to extract a relationship, we should first traverse our knowledge base to get the frequency of this relation word. If the frequency number is greater than 500, the corresponding relation is high-frequency; otherwise we regard it as low-frequency relation. The following two sections will introduce different methods to extract these two kinds of relation.

3.2 Candidate Sentences Selecting Method

When the relation word is high-frequency in our knowledge base, we will traverse the knowledge base to get the corresponding $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$ triples. And these triples are used to locate candidate sentences in wiki-page of Baidu encyclopedia and Hu-dong encyclopedia corpus to extract this relation. The sentences chose as candidate sentences contain information for the relation word. For example, a sentence is “李安导演的《卧虎藏龙》诠释了中国武侠的魅力”, and the relation word we concerned is “导演”, and then this sentence will be selected as candidate sentence.

Here, four different approaches are proposed to estimate whether a sentence is a candidate sentence for a triple of the target relation word.

We explore two methods to score a sentence:

$$\text{score} = b\text{Arg} 1 * (b \text{Re} l + 1) * b\text{Arg} 2 \quad (1)$$

$$\text{score} = (b\text{Arg} 1 + 1) * (b \text{Re} l + 2) * b\text{Arg} 2 \quad (2)$$

We define three variables: $b\text{Arg}1$, $b\text{Rel}$, $b\text{Agr}2$. For a triple $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$ and a sentence, if the arg1 or its alias name appears in the sentence, $b\text{Arg}1 = 1$, otherwise $b\text{Arg}1 = 0$; if the rel or its alias name appears in the sentence, $b\text{Rel} = 1$, otherwise $b\text{Rel} = 0$; if the arg2 or its alias name appears in the sentence, $b\text{Arg}2 = 1$, otherwise $b\text{Arg}2 = 0$.

We choose the sentence of the highest score and the score must be greater than 0. Obviously, the sentence gaining by the first scoring method must have the arg1 and the arg2 , and the sentence obtaining by the second method only must have the arg2 .

Most of the time, there are many sentences with same highest score. Here, we propose two methods to get the final candidate sentences from these highest score sentences:

- (a) Selecting the highest score sentence first appeared in an article;
- (b) Selecting all highest score sentences.

After combination, we have four methods to obtain the candidate sentences, and then these sentences are used as training data.

Then we want to extract triples from the wiki-page content and the corresponding entity should not be in our semantic knowledge base. One simple idea is segmenting the article into sentences, and then extracts entity relation triples from all these sentences. Of course, there will be too many sentences need to extract, and we can reduce the candidate sentences.

Here, we introduce another method of getting less candidate sentences for extracting. Firstly, we will do word segmentation and part-of-speech tagging for the candidate sentences which are chose as training data, and then choose the nouns and verbs. Secondly, we do word frequency statistic for the selected word, and choose the top-n highest frequency words as key words. At last, these key words are used to determine the candidate sentences for extracting, and these candidate sentences are used as testing data.

For a triple $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$, the rel is the relation word we concerned, and the arg1 is the entity of the wiki-page, and then we only need to get the arg2. So the Chinese entity relation extraction is converted to annotation problem. We train conditional random field model to label the arg2 in the testing data, and finally convert the annotation results to entity relation triples.

3.3 The Heuristic Rules Based Method

If a relation word is low-frequency in our knowledge base, we can't automatically get enough training data for statistical models. For this reason, we propose the heuristic rules based entity relation extraction algorithm, as shown in the following.

Algorithm1: The Heuristic Rules based Entity Relation Extraction Algorithm

Input: The target relations, some entities, corresponding categories and unstructured content

Output: Entity relation triples $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$, arg1 and arg2 are entities and rel is the relation

- 1 Begin
 - 2 Confirm the template $\langle \text{class1}, \text{rel}, \text{class2} \rangle$ of a target relation; here class1 and class2 are the categories of unknown entities. For example, a given relation “director”, we can confirm class1 is movie or teleplay and class2 is people.
 - 3 Produce an entity library, which contain entities and corresponding categories.
 - 4 Get the keywords of target relation by using domain knowledge.
 - 5 Select candidate sentences, which should contain keywords, and entities of class1 and class2 in our entity library.
 - 6 Generate some simple rules to extract the entity relation.
 - 7 End
-

Here we briefly explain the steps in this algorithm, combined with the competition of Sougou Chinese entity relation extraction. This competition involves 5 categories and a total of 17 relationships for extracting. And the movie category is similar with the teleplay category, and then we only need to discuss 4 categories and 12 relations. Some relation words are high-frequency and some are low-frequency, as shown in table 1. We adopt the method introduced in Section 3.2 to extract the high-frequency relation, and our heuristic rules based method is used to extract low-frequency relations.

Table 1. Categories and relation words of Sougou entity relation extraction competition

Frequency	Relation Words	Category
High-frequency	导演(director),演员(actor),编剧(writer),	Movie/ Teleplay
	作者(writer)	Book
Low-frequency	演唱者(singer), 作词(writer), 作曲(composer)	Song
	父母 (parent), 兄弟姐妹 (brother or sister), 夫妻 (husband or wife)	People
	原著(original book), 原创音乐(Music soundtrack)	Movie/ Teleplay

The relation words which we concerned in people category are : “父母”, “兄弟姐妹” and “夫妻”. So we can confirm that class1 and class2 are both people category. We build an entity library containing all given people entities. The keywords for the relation word “父母” are : “父”, “爹”, “爸”, “子”, “女”, “母” and “妈”. The keywords for the relation word “夫妻” are : “结婚”, “完婚”, “闪婚”, “老公”, “老婆”, “妻子”, “丈夫”, “妻”, “夫”, “娶” and “嫁”. The keywords for the relation word “兄弟姐妹” are : “兄”, “哥”, “弟”, “姐”, “妹” and “姊”. Then we should select the sentences containing at least two people entities, and the sentences should have the keywords for different relation extraction.

When extracting the relation word “父母”, if the keywords are: “父”, “爹”, “爸”, “母” or “妈”, the entity before these keywords is the arg1 in triple <arg1, rel, arg2>, and the entity after these keywords is the arg2. But for the keywords of “子” and “女”, the entity before these keywords is the arg2, and the entity after these keywords is the arg1. When extracting the relation word of “兄弟姐妹” and “夫妻”, we don't need to consider the order. We can get entity triples <arg1, rel, arg2> by using these simple rules.

We adopt the heuristic rules based entity relation extraction algorithm for the rest low-frequency relation words, and the detailed steps is similar to method described above.

4 Experimental Results and Analysis

We adopt the improved candidate sentences selecting method trained by conditional random field model to extract the high-frequency relation; section 4.1 will introduce the comparison and analysis of our experiment results. And section 4.2 will introduce our results for the competition of Sougou Chinese entity relation extraction.

4.1 The Comparison of Various Methods of Select Candidate Sentences

We select 5 categories and 2 relation words of every category, listed in Table 2.

Table 2. Five categories and two relation words of every category for extraction

Category	Relation Words
Geography	area, district
Movie	writer, director
Education	starting time, category
Book	publishing company, publication time
People	height, weight

In preparing for training data step, there are two methods to score a sentence based on the given triple $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$:

(a) Score method 1: $score\ 1 = bArg1 * (bRel + 1) * bArg2$

(b) Score method 2: $score\ 2 = (bArg1 + 1) * (bRel + 2) * bArg2$

- If arg1 appears in this sentence, then $bArg1 = 1$, otherwise $bArg1 = 0$.
- If arg2 appears in this sentence, then $bArg2 = 1$, otherwise $bArg2 = 0$.
- If rel appears in this sentence, then $bRel = 1$, otherwise $bRel = 0$.

In preparing for training data step, two methods to get the final candidate sentences from these highest score sentences:

- (a) Selecting the highest score sentence first appeared in an article;
- (b) Selecting all highest score sentences.

In preparing the data for extracting step, two methods to extract triples from the wiki-page content (testing data):

- (a) Choosing all the sentences in the wiki-page content;
- (b) Selecting some sentences from the wiki-page content based on keyword matching.

Annotation:

- Label 1: score a sentence by score method 1, and then selecting all highest score sentences.
 - Label 2: score a sentence by score method 1, and then selecting the highest score sentence first appeared in an article.
 - Label 3: score a sentence by score method 2, and then selecting all highest score sentences.
 - Label 4: score a sentence by score method 2, and then selecting the highest score sentence first appeared in an article.
- (1) Choosing all the sentences in the wiki-page content, part of the experiment results:

Table 3. The extraction results of choosing all the sentences in the wiki-page content

Relation Words	Precision	Recall	F1	Relation Words	Precision	Recall	F1
Geo_area1	0.1570	0.1371	0.1463	Movie_Director1	0.1443	0.0833	0.1057
Geo_area2	0.1600	0.1421	0.1505	Movie_Director2	0.1633	0.0952	0.1203
Geo_area3	0.1486	0.1320	0.1398	Movie_Director3	0.1782	0.1071	0.1338
Geo_area4	0.1534	0.1371	0.1448	Movie_Director4	0.1458	0.0833	0.1061
Geo_district1	0.3118	0.2944	0.3209	EDU_Start_time1	0.3097	0.2909	0.3000
Geo_district2	0.3059	0.2640	0.2834	EDU_Start_time2	0.3117	0.2909	0.3009
Geo_district3	0.3333	0.3147	0.3238	EDU_Start_time3	0.3397	0.3212	0.3302
Ge_district4	0.3086	0.2741	0.2903	EDU_Start_time4	0.3333	0.3152	0.3240

(2) Selecting some sentences from the wiki-page content based on keyword matching, part of the experiment results:

Table 4. The extraction results of selecting some sentences from the wiki-page content based on keyword matching

Relation Words	Precision	Recall	F1	Relation Words	Precision	Recall	F1
Geo_area1	0.3119	0.1726	0.2222	Movie_Director1	0.4833	0.1726	0.2544
Geo_area2	0.3736	0.1726	0.2361	Movie_Director2	0.5439	0.1848	0.2756
Geo_area3	0.2661	0.1675	0.2056	Movie_Director3	0.4110	0.1786	0.2490
Geo_area4	0.2623	0.1624	0.2006	Movie_Director4	0.5469	0.2083	0.3017
Geo_district1	0.4294	0.3706	0.3978	EDU_Start_time1	0.6993	0.6061	0.6494
Geo_district2	0.4000	0.2538	0.3106	EDU_Start_time2	0.7252	0.5758	0.6419
Geo_district3	0.4535	0.3959	0.4228	EDU_Start_time3	0.7329	0.6485	0.6881
Ge_district4	0.4031	0.2640	0.3190	EDU_Start_time4	0.7211	0.6424	0.6795

Table 3 and Table 4 show the effect of different options to the extraction results, and after comparison and analysis we can get the follow conclusion:

1, the real results should be better than shown above, because we use the number of web pages instead of the number of triples that need be extracted, and in fact some web pages don't have triples, so the number of triples that need be extracted is less than the number of web pages, and the actual recall rate will be higher.

2, the extraction results of different relation words vary a lot, some results are very good, but some are not. We can easy find this

3, in the annotation, four different labels represent four different methods to get candidate sentences to train the conditional random field model. After comparing the results, we can find that the method of label 2 can get the highest precision and the

method of label 3 can get the highest recall, and it is hard to conclude which method can get the highest F1 value. The method of label 2 can get accurate and related training data, so this method can achieve the highest precision. The method of label 3 can get abundant training data to achieve the highest recall.

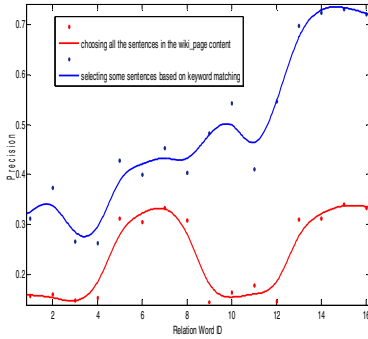


Fig. 2. The precision of different candidate sentences selecting methods

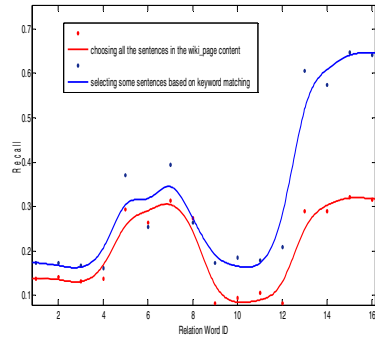


Fig. 3. The recall of different candidate sentences selecting methods

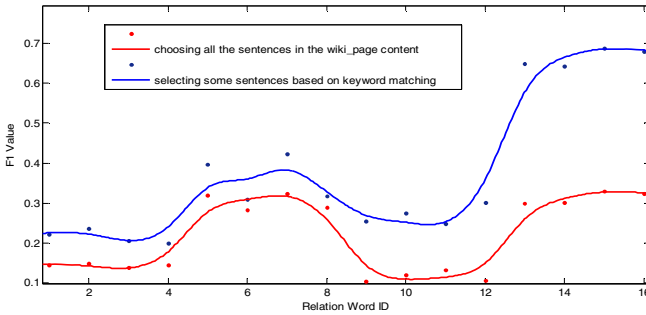


Fig. 4. The F1 Value of different candidate sentences selecting methods

When we apply the method of selecting some sentences based on keywords matching, the precision and recall have improved significantly (see Fig 2 and Fig 3). And the candidate sentences selecting method improve the average F1 value by 78.53%, and from Fig 4 we can find the F1 value increased obviously. So selecting candidate sentences is necessary, and our method has a good effect.

4.2 The Competition of Sougou Web-Based Entity Relation Extraction

The Sougou Company has their knowledge base of entity relation triples, and there is a test platform based on these triples for competitors to verify their results. But the results of test platform are not the final results, and the final results need sampling and

artificial validation to determine, and the organizing committee will give the final results later.

Table 5. Some example of Sougou Web-based Entity Relation Extraction Competition

Category	Relation	Sentence	Triples
人物	父母	冉甲男与父亲冉平一起担任编剧的电影《画皮2》备受期待。	<冉甲男, 父母, 冉平>
	夫妻	林姮怡与蒋家第四代蒋友柏结婚, 婚后息影。	<林姮怡, 夫妻, 蒋友柏>
	兄弟姐妹	曾维信的奶奶胡菊花, 是胡耀邦的亲姐姐。	<胡菊花, 兄弟姐妹, 胡耀邦>
书籍	作者	《沙床》当代高校生活的青春怦怦情录作者: 葛红兵。	<沙床, 作者, 葛红兵>
歌曲	作词	《幻想爱》是陈伟作词作曲, 张韶涵演唱的一首歌曲。	<幻想爱, 作词, 陈伟>
	作曲		<幻想爱, 作曲, 陈伟>
	演唱者		<幻想爱, 演唱者, 张韶涵>
电影/电视剧	导演	李安导演的《卧虎藏龙》诠释了中国武侠的魅力。	<卧虎藏龙, 导演, 李安>
	编剧	电影海上烟云由柯枫自编自导。	<海上烟云, 编剧, 柯枫>
	原著	根据琼瑶原著《含羞草》改编的台湾电视连续剧《含羞草》。	<含羞草, 原著, 含羞草>
	演员	电视剧《龙堂》由著名演员张丰毅、陈小春主演。	<龙堂, 演员, 张丰毅> <龙堂, 演员, 陈小春>
	原声音乐	电影《大兵金宝历险记》主题曲是刘佳演唱的美丽国。	<大兵金宝历险记, 原声音乐, 美丽国>

Table 5 shows some extraction results for the competition. This competition involves 5 categories and a total of 17 relationships for extracting. And the movie category is similar with the teleplay category, so we combine these two categories. Then we only need to discuss 4 categories and 12 relations.

Finally we submitted a total of 364944 triples. The precision is 49.09% and we rank the fourth place.

5 Conclusion and Future Work

In this paper, we first build a knowledge base using the collected corpus, and then adopt different methods to get the <arg1, rel, arg2> triples for different frequency relation words of our knowledge base. In the experiment, a detailed comparison and analysis on some options of selecting candidate sentences are introduced, and then we participate in the competition of Sougou Chinese entity relation extraction and rank the fourth place in the test platform.

However, in our experiment, only lexical features are used for triples extraction, and the parser-based features are ignored. An interesting direction is how to combine the parser-based features into the previous work, and in turn improving the performance of our extraction work.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (NSFC) Grants No.61203281 and No.61303172.

References

1. Message Understanding Conference,
http://en.wikipedia.org/wiki/Message_Understanding_Conference
2. Automatic Content Extraction,
<http://www.itl.nist.gov/iad/mig/tests/ace/>
3. Etzioni, O., Banko, M., Soderland, S., et al.: Open information extraction from the web. *Communications of the ACM* 51(12), 68–74 (2008)
4. Yates, A., Cafarella, M., Banko, M., et al.: TextRunner: open information extraction on the web. In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 25–26. Association for Computational Linguistics (2007)
5. Wu, F., Weld, D.S.: Open information extraction using Wikipedia. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 118–127. Association for Computational Linguistics (2010)
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545. Association for Computational Linguistics (2011)
7. Etzioni, O., Fader, A., Christensen, J.: Open information extraction: The second generation. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 3–10. AAAI Press (2011)
8. Schmitz, M., Bart, R., Soderland, S., et al.: Open language learning for information extraction. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523–534. Association for Computational Linguistics (2012)
9. Zhang, Y., Zhou, J.: A trainable method for extracting Chinese entity names and their relations. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, vol. 12 (2000)
10. Zeng, Y., Wang, D., Zhang, T., Wang, H., Hao, H.: Linking Entities in Short Texts Based on a Chinese Semantic Knowledge Base. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) *NLPC 2013. CCIS*, vol. 400, pp. 266–276. Springer, Heidelberg (2013)
11. Chen, Y., Chen, L., Xu, K.: Learning Chinese Entity Attributes from Online Encyclopedia. In: Wang, H., et al. (eds.) *APWeb Workshops 2012. LNCS*, vol. 7234, pp. 179–186. Springer, Heidelberg (2012)

Linking Entities in Tweets to Wikipedia Knowledge Base

Xianqi Zou, Chengjie Sun, Yaming Sun, Bingquan Liu, and Lei Lin

School of Computer Science and Technology
Harbin Institute of Technology, China

{xqzou, cjsun, ymsun, liubq, linl}@insun.hit.edu.cn

Abstract.: Entity linking has received much more attention. The purpose of entity linking is to link the mentions in the text to the corresponding entities in the knowledge base. Most work of entity linking is aiming at long texts, such as BBS or blog. Microblog as a new kind of social platform, entity linking in which will face many problems. In this paper, we divide the entity linking task into two parts. The first part is entity candidates' generation and feature extraction. We use Wikipedia articles information to generate enough entity candidates, and as far as possible eliminate ambiguity candidates to get higher coverage and less quantity. In terms of feature, we adopt belief propagation, which is based on the topic distribution, to get global feature. The experiment results show that our method achieves better performance than that based on common links. When combining global features with local features, the performance will be obviously improved. The second part is entity candidates ranking. Traditional learning to rank methods have been widely used in entity linking task. However, entity linking does not consider the ranking order of non-target entities. Thus, we utilize a boosting algorithm of non-ranking method to predict the target entity, which leads to 77.48% accuracy.

Keywords: entity linking, global feature, topic distribution, boosting algorithm.

1 Introduction

Entity linking, a task to make the mentions certain in tweets, is to link mentions to the unambiguous knowledge base. Entity linking usually can be divided into two major steps: entity candidates' generation and ranking.

The first step is to generate candidate sets of entities. The entity candidates' generation process often involves query expansion. Mining the text of mentions and making full use of various resources will expand mentions to the entities that are literally close. Using search engine with the HTML marks could obtain huge sets of relevant candidates. The main drawback is that this approach requires to crawl a lot of documents from the search engine. For different search engines, we need to make different rules to expand the mentions. We utilize Wikipedia article pages as extended entity reference resources and fully mine the Wikipedia article's text. To a specified mention, the entity candidates will be quite large, and include many ambiguous candidates. Therefore, we design a reasonable classifier to reduce the number of entity candidates and increase the coverage of target entities.

As the language used in tweet is not formal and the text is short, only limited features could be effectively extracted. For entity linking, features are usually represented in space vector. The element's value in the vector can be binary or TF-IDF, and this vector is very sparse. The local features mainly focus on the relationships between entities and mentions. In addition to the context semantic similarity of TF-IDF, there are anchor text prior probability, string edit distance between mention and entity's Wikipedia article title, the position of ranking returned by search engine, the length and page views of the candidates' Wikipedia articles. Global features are mainly considering the relationship among candidates. For different mentions from the same tweet, their target entities should have coordinate relationship. We use topic distribution instead of common links to get global feature during belief propagation. Topic distribution provides more information than common links in the perspective of semantics.

A problem often faced in information retrieval is to rank query's related documents, in the recommendation system is primarily to recommend interested items to users, however in entity linking is to rank the entity candidates and choose the candidate of the Top1. Ranking task usually adopts scoring mechanism. However, in entity linking, learning to rank method is often used to rank the candidates. Pairwise approach will transform the order of entity candidates into pairs, which the more related entity with the less one to be labeled as +1, otherwise -1. While the listwise approach makes use of the order sequence of entity candidates. In information retrieval, the number of training set is very large due to the fact that every pair of the candidates and every different order sequence could be made into a new training example. Since only predicting the Top1 candidate as the result of candidates' ranking without considering the other candidates' relative ranking and the mentions in entity linking have the similar feature distribution, we believe the model based on single candidate will produce a better result in entity linking task. We adopt Regularized Greedy Forest model [1], which is based on Gradient Boosting Decision Tree [2], to predict the target candidate. Our experiment results show that this method which doesn't rely on relative ranking features could produce a better performance.

2 Related Work

Entity candidates' generation directly affects the performance of the entity linking. Generating entity candidates usually adopts abbreviated words expansion, domain dictionary building. Chang [3] used online abbreviation dictionary to expand relatively formal medical abbreviations. Han [4] matched the words before the phrases, such as "also called", "known as", "short for". But these methods have not considered abbreviations for informal words. Nadeau [5] used supervised machine learning methods and taken advantage of part of speech information. However, this method could only be limited to the situation that abbreviation and the expansion of words are in the same sentence. When extended to the other documents, the performance would be seriously affected by the noise data. Zhang [6] also made use of supervised machine learning methods which relied not only on HTML mark and language clues, but also on semantic information.

Features measuring mention and entity’s relevancy are referred to local features. Local features often used vector dot product, cosine similarity, K-L divergence, Jac-card distance, etc. Liu [7] utilized both local features and global features. Local features included edit distance similarity between mentions and entities, the probability of mention as anchor text to be appeared in the entity’s Wikipedia article, etc. Global features described the relationship among the entities of different mentions in a tweet, such as the common link of entities’ Wikipedia articles. Han [8] built a collective graph model to coordinate the relationship among entities of different mentions and correlate mentions with entities in Wikipedia knowledge base. Zheng [9] taken advantage of deep neural network to represent the mentions’ text and entities’ Wikipedia articles respectively.

Linking mentions into knowledge base is usually seen as a way to provide semantic information. The idea has been widely applied to various kinds of media, such as text, multimedia, news, radio reports, etc. Milene and Witten [10] used traditional classifiers such as Naïve Bayes, C4.5[11]. Meij adopted the Gradient Boosting Decision Tree model to predict the target entity. Another common method to link mentions to entities in knowledge base was to use learning to rank model. Zheng [12] utilized pairwise method (Ranking Perceptron [13]) and listwise method (ListNet [14]). In this paper, we make use of belief propagation process on the topic distribution to obtain global feature and analyze the validity of this feature and other local features. Then we compare the learning to rank method with Regularized Greedy Forest, an improved Gradient Boosting Decision Tree model.

3 Candidates Generation and Ranking

3.1 Candidates Generation

Entity candidates’ generating process involves query expansion. Various resources could be made full use of to expand the mentions. We mine the Wikipedia pages and extract redirect page title, disambiguation page title, bold words in first paragraph,

Table 1. Feature set for filtering ambiguous candidates

Feature	Feature Description
Cap_All	Whether mention is capitalized.
LCS_First	Edit distance between mention and first letter of entity word.
Length_3	Whether the length of mention is less than 3.
Parenthesis	Whether entity has parenthesis.
Match_All	Whether mention and entity’s string match exactly.
Redirects	Whether entity is from redirect page.
Disam	Whether entity is from disambiguation page.
Anchor	Whether entity is anchor text.
Bold	Whether entity is bold.

anchor texts of the mention's candidates. The set of entity candidates is quite large and contains many ambiguous candidates. Thus we train SVM model to filter parts of the candidates to ensure the quality of the entity candidates. Table 1 shows the features we used to train our model.

3.2 Candidates Ranking

Features for Linking

In this part, we will introduce the features used for candidates ranking. The features include local features and global features. First we define the symbol used in describing features as showed in Table 2.

Table 2. Symbol used to describe feature

Symbol	Description
M	Mention.
E	Entity candidate.
E_i	Entity candidate of the i th mention.

- f_1 is the prior probability which is the quantitative proportion of M to be presented as anchor text in M's candidates' Wikipedia article.
- f_2 is edit distance similarity between E and M.
- f_3, f_4 are binary values. The feature value will be 1, if M contains E, otherwise 0.
- f_5 is a feature related with ranking position of Wikipedia search result. If the position is in the first place, the value will be 1, otherwise 0.
- f_6 is the TF-IDF similarity between M's tweet and E's Wikipedia article.
- f_7, f_8 are E's Wikipedia article's length and page view counts respectively.
- f_9, f_{10} are links similarity and topic distribution similarity between E_i and E_j of different mentions in a tweet.

Wikipedia articles contain a large number of anchor texts. These text information together with the non-text information, such as text length and page view counts, can be viewed as the prior knowledge. Feature f_9, f_{10} are global features which describe the coordination degree between two entities of different mentions while the others are local features which only consider the relationship between mention and its candidate entity.

If the entity candidates of different mentions are related to some extent, their Wikipedia articles will also have common inner links. The more the same common inner links they have, the more similar their contents are. The candidates of different mentions in a tweet also have semantic relationships to some degree. The same topic distribution should be followed by these entity candidates. We take advantage of belief

propagation on the candidates’ common links and topic distribution to obtain the coordination degree between mentions and their candidates. The belief propagation is showed as:

$$V_k = (1 - \lambda) * B * V_{k-1} + \lambda * V_0 \tag{1}$$

In the belief propagation, we use V to represent the belief vector. The vector has n_1 elements related with mentions and n_2 elements with candidates. B is the matrix of belief with $(n_1 + n_2)^2$ elements. $B[i, j]$ is the belief that element j propagates to element i . If element j is an entity and element i is a mention, the value of $B[i, j]$ is showed in formula 2. If element j is an entity and element i is also an entity, $B[i, j]$ will use formula 3 as its value. Otherwise, $B[i, j]$ will be 0. We initial the belief vector $V_0[1 : n_1]$ with formula 2. $V_0[n_1 + 1 : n_1 + n_2]$ is set to be 0. λ is the tradeoff between original and reallocated belief. After the K times of iterations in the belief propagation, we will get the final belief vector V_k .

$$P(E | M) = \frac{TFIDF(M, E)}{\sum_{E \in C_M} TFIDF(M, E)} \tag{2}$$

$$P(E_j | E_i) = \frac{S(E_i, E_j)}{\sum_{E \in N_{E_i}} S(E_i, E)} \tag{3}$$

Where $S(E_i, E_j)$ could make use of various similarity measurement between two candidates of different mentions in a tweet. The similarity between two candidates which is based on common links could be obtained as:

$$LS(E_i, E_j) = 1 - \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))} \tag{4}$$

Where W is the counts of Wikipedia articles and A, B are the sets of entities which link to entity E_i, E_j respectively. In this paper, we make use of topic distribution similarity to compute $S(E_i, E_j)$.

Methods for Linking

Ranking problem in entity linking and information retrieval have the same substance. Thus the method of learning to rank is often used to rank the entity candidates from pairwise approach(RankSVM [15], RankBoost [16], RankNet [17]) to listwise approach(AdaRank [18], ListNet). We take advantage of Regularized Greedy Forest. According to Friedman, when setting the shrinkage parameter small enough, the gradient boosting could achieve good performance. The Fully-Corrective Gradient Boosting is to avoid the small step size problem as showed in Algorithm 1.

Algorithm 1. Fully-Corrective Gradient Boosting

```

 $h_0(x) \leftarrow \arg \min_{\rho} L(\rho, Y)$ 
for  $k = 1$  to  $K$  do
   $\tilde{Y}_k \leftarrow - \left[ \frac{\partial L(h, Y)}{\partial h} \right]_{h=h_{k-1}(X)}$ 

   $b_k \leftarrow A(X, \tilde{Y}_k)$ 
  let  $H_k = \left\{ \sum_{j=1}^k \beta_j b_j(x) : \beta_j \in \mathbb{R} \right\}$ 
   $h_k \leftarrow \arg \min_{h \in H_k} L(h(X), Y)$ 
end
return  $h(x) = h_K(x)$ 

```

Regularized Greedy Forest uses the Fully-Corrective Gradient Boosting to learn a decision forest. At the same time, all the parameters of the trained decision trees would be adjusted during each model training time, such as the number of decision trees, leaf nodes. The sparse combination of decision rules is completed by the greedy search. After obtaining a large number of decision trees, RGF will regularized the model appropriately.

4 Experiments

4.1 Experiment Setting

We adopt the Yahoo scientist Meij’s annotation data set. We divide 502 tweets into 5:1 as the training set and testing set. For the text information and non-text information related with entities, we make use of the Wikipedia API to get them. The metrics to measure the experiment’s results are

$$\text{Accuracy} = \frac{|\{C_{i,0} \mid C_{i,0} = \zeta_i\}|}{N} \quad (5)$$

$$\text{Coverage} = \frac{|\{\zeta_i \mid \zeta_i \in C_i\}|}{N} \quad (6)$$

Where C_i is the candidate set of mention i , $C_{i,0}$ is the candidate which ranks at the first position for mention i . ζ_i is the gold standard annotation for mention i . N is the number of mentions in the data set.

Firstly, we make an experiment to decide the size of candidates set which to ensure a higher coverage of the gold candidate. Secondly, we verify the effectiveness of local features and global features respectively. In the end, we compare the learning to rank methods with Regularized Greedy Forest.

4.2 Results and Analysis

We train a SVM classifier with all the features described in Table 1 to make a binary classification of all entity candidate. For the positive candidates our model predicts, we utilize the edit distance between mention and the entity candidate to choose the Top K of them. Different value of K will effects the coverage of target candidate. As is shown in Fig. 1, if K is 1, the coverage of target entity will be 64%. When we set K to be 20, the target entity coverage no longer grows. Therefore, we make use of the Top 20 candidates of the SVM classifier result as the candidates.

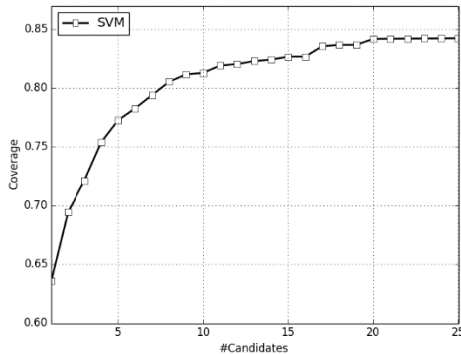


Fig. 1. Coverage of different size of candidates set

We extract local features, global features and utilize a basic model (RankSVM) to analyze the validity of them. Local features contain text information and non-text information. Table 3 shows that feature f_5 is the most effective among these local features. Since feature f_5 is associated with the Wikipedia search results, it has the effect of reducing ambiguity partly. Feature f_2, f_3, f_4 are mainly considering the literal relationship between mention and candidate. Compared with feature f_5 , their accuracy is relatively lower. Feature f_8 , as a prior knowledge, achieves 62.16% accuracy.

Table 3. Local Feature Analysis

Local Feature	Accuracy(%)
f_1	17.12
f_2	59.46
f_3	64.86
f_4	54.05
f_5	65.77
f_6	33.33
f_7	9.91
f_8	62.16

For global features, feature f_{10} is more effective than feature f_9 as showed in Table 4, since feature f_{10} could provide more latent semantic information than feature f_9 . When local features combined with global features, the accuracy has reached to 72.97% in Table 5. The convergence rate of the belief propagation progress based on different entity similarity is also different. Fig. 2 shows that the convergence rate on feature f_{10} is faster to feature f_9 . The entities that have the same inner links may be about different topics in the Wikipedia page. Since feature f_9 has more noise information, its belief propagation iterates more steps to get a balance state.

Table 4. Global Feature Analysis

Global Feature	Accuracy(%)
f_9	9.01
f_{10}	11.71

Table 5. Performance of Commbined Features

Local Feature + Global Feature	Accuracy(%)
$f_1+f_2+f_3+f_4+f_5+f_6+f_7+f_8$	70.27
f_9+f_{10}	13.51
$f_1+f_2+f_3+f_4+f_5+f_6+f_7+f_8+f_9+f_{10}$	72.97

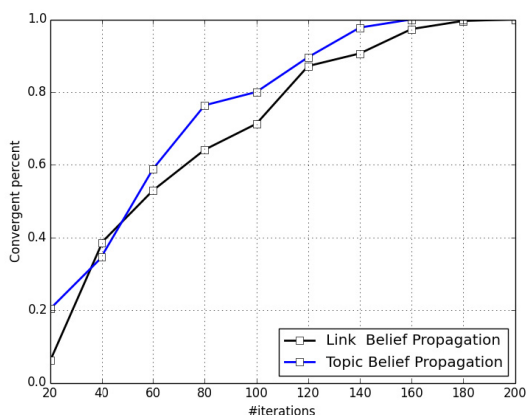


Fig. 2. Convergent Percent of Tweets during the Belief Propagation Based on Different Entity Similarity

Table 6 has shown that the performance of pairwise approach is better than listwise approach. As in entity linking task, the rank position of target entity is +1 while non-target entities are -1 equally. Therefore, the listwise approach relies more on single feature to make the probability of one ranking combination higher while others lower

and equal. The learning to ranking methods lose more information about non-target entities. Compared learning to rank methods with Regularized Greedy Forest, the later has achieved a higher accuracy with the accuracy of 77.48%. In tweets, the candidates' features distribution of different mentions is similar, while in information retrieval, the documents' feature distribution of different queries may vary a lot. Thus the RGF model, which trained on the original data, could achieve a better performance.

Table 6. Experiment results comparison of different methods

Model	Accuracy(%)
RankSVM	72.97
RankBoost	72.07
RankNet	65.77
AdaRank	67.57
ListNet	66.67
GBDT	74.77
RGF_L2	77.48

Fig. 3 shows the accuracy of different leaves during the training process of RGF model. Using L2 regularization with the leaves, RGF model could achieve the highest accuracy when the number of leaves is 1000.

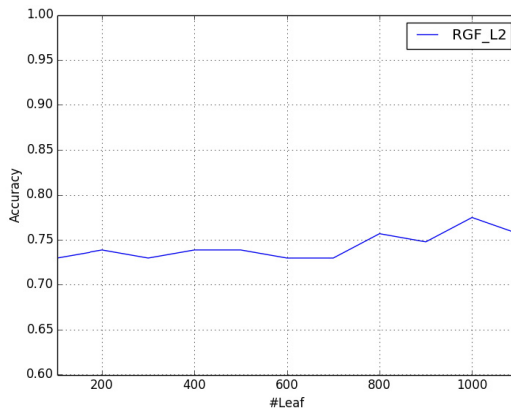


Fig. 3. The Accuracy of Different Leaves in RGF_L2

5 Conclusions

In this paper, we adopt the method of belief propagation based on the topic distribution to obtain global feature, which achieves better performance than that uses

common links. When combining global features with local features, performance will be obviously improved. We compared the traditional learning to rank methods with Regularized Greedy Forest. Experiment results show that the RGF model could achieve a better performance in the tweets' entity linking task.

Acknowledgment. This work is supported by the Key Basic Research Foundation of Shenzhen (JC201005260118A) and National Natural Science Foundation of China (61100094 & 61300114). The authors are grateful to the anonymous re-viewers for their constructive comments.

References

1. Johnson, R., Zhang, T.: Learning nonlinear functions using regularized greedy forest. arXiv preprint arXiv:1109.0887 (2011)
2. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232 (2001)
3. Chang, J.T., Schütze, H., Altman, R.B.: Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Informatics Association* 9(6), 612–620 (2002)
4. Han, X., Zhao, J.: NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. In: *Proceedings of Text Analysis Conference (TAC 2009)* (2009)
5. Nadeau, D., Turney, P.: A supervised learning approach to acronym identification (2005)
6. Zhang, W., Sim, Y.C., Su, J., Tan, C.L.: Entity linking with effective acronym expansion, instance selection and topic modeling. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, vol. 3, pp. 1909–1914 (2011)
7. Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., Lu, Y.: Entity linking for tweets. In: *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (2013)
8. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: A graph-based method. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 765–774 (2011)
9. He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., Wang, H.: Learning entity representation for entity disambiguation. In: *Proc. ACL 2013* (2013)
10. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 509–518 (2008)
11. Quinlan, J.R.: *C4. 5: programs for machine learning*. Morgan Kaufmann (1993)
12. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 483–491 (2010)
13. Shen, L., Joshi, A.K.: Ranking and reranking with perceptron. *Machine Learning* 60(1-3), 73–96 (2005)
14. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to rank: From pairwise approach to listwise approach. In: *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136 (2007)
15. Joachims, T.: Optimizing search engines using clickthrough data. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 133–142 (2002)

16. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research* 4, 933–969 (2003)
17. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96 (2005)
18. Xu, J., Li, H.: Adarank: a boosting algorithm for information retrieval. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 391–398 (2007)

Automatic Recognition of Chinese Location Entity

Xuewei Li¹, Xueqiang Lv¹, and Kehui Liu^{2,3}

¹ Beijing Key Laboratory of Internet Culture and Digital Dissemination Research,
Beijing Information Science and Technology University, Beijing, China

li_xuewei163@163.com lxq@bistu.edu.cn

² Beijing Institute of Technology, Beijing, China

³ Beijing Research Center of Urban Systems Engineering, Beijing, China

lkh_2005@126.com

Abstract. Recognition of Chinese location entity is an important part of event extraction. In this paper we propose a novel method to identify Chinese location entity based on the divide-and-conquer strategy. Firstly, we use CRF role labeling to identify the basic place name. Secondly, by using semi-automatic way, we build indicator lexicon. Finally, we propose attachment connection algorithm to connect the basic place name with indicator, then we achieve the identification of location entity. In brief, our method decomposes location entity into basic place name and indicator, which is different from traditional methods. Results of the experiments show that the proposed method has an outstanding effect and the F-value gets to 84.79%.

Keywords: Chinese location entity. Divide-and-conquer strategy. CRF role labeling. Basic place name. Indicator lexicon. Attachment Connection Algorithm.

1 Introduction

Urban management enters into the age of information and people raise issues about urban management through the Internet. Through non-standard writing, texts of complaint format are quite different. Thus, the staff must read verbatim to find important event from exponential growth of texts of complaint about urban management that is shown in table 1, which is inefficient.

By adopting the technique of information extraction, we can extract event automatically by converting the unstructured data into structured data. It not only improves the work efficiency, but also helps the urban management department to master the implementation of policy and find the existing problems in the social management. Besides, automatic recognition of location entity, which is shown as italic and underline in table 1, is an important part of event extraction.

Table 1. The texts of complaint about urban management and location entities examples

Experimental Corpus and location entities example (italic and underline indicates location entities)	
Text 1	<p>标题：<u>关于马家堡西路角门西地铁站外面的丁字路口</u>的问题</p> <p>来信内容：<u>1.马家堡西路角门西地铁站外面的丁字路口，南北向人行横道上北面</u>横着一道30米左右的护栏。……望有关部门早日解决以上问题</p> <p>标题：整治环境</p>
Text 2	<p>来信内容：<u>海淀区西四环定慧北桥以东的定慧福里小区南面的停车场</u>简直就是个垃圾站，……。<u>定慧福里北面及家乐福前面的道路上</u>，……。如：<u>从定慧寺车站到定慧桥车站路的北面人行道上</u>经常有狗屎。……北京是首都，因此也会影响到中国在世界上的形象。</p> <p>标题：<u>海淀区黑泉路一个井盖</u>缺失</p>
Text 3	<p>来信内容：……地点在<u>黑泉路南段，北向南方向非机动车道，北五环林萃桥北200米左右</u>。</p>

As can be seen from table 1, location entity is made up of basic place name and indicator. Similarly, the divide-and-conquer strategy can decompose complex problem into smaller parts and solve them. In this paper, we borrow the idea of divide-and-conquer strategy to divide location entity recognition into basic place name recognition and indicator lexicon construction. Then attachment connection algorithm was put forward, which is ACA, to connect basic place name with indicator.

The contributions of this paper are twofold. Firstly, the research on location entity recognition in the text of complaint about urban management has never appeared before. Secondly, we propose a model that integrates divide and conquer strategy in location entity recognition, which is quite different from other ones. To the best of our knowledge, this paper is the first one which has introduced indicator in location entity. As shown in the experiment section, our method gains reasonable performance.

The rest of the paper is organized as follows. Some related works are discussed in section 2. Then we will introduce some basic concepts as basic place name, indicator, as well as location entity and present our model in section 3. After that, experiment results and discussions are showed in section 4. Finally, section 5 concludes the whole paper and put forward some work to be done in the future.

2 Related Work

Currently, the domestic related studies on recognition of Chinese place name mainly focus on texts which the format is standard. Cai et al. [1] proposed rule reasoning based method to identify unexpected event place name entity from news corpus and extract place name entities including province, city, county, township and village by analyzing

expressive features of unexpected event place name entity. Li et al. [2] defined it as a binary classification problem and applied a SVM classifier to identify Chinese place names with key words, such as province and city, from People's Daily. Tang et al. [3] employed CRFs-based module for simple location name recognition from People's Daily corpus. Du et al. [4] recognized Chinese place names in news page corpora. Other scholars [5,6,7,8,9] also studied the recognition of simple Chinese place names in standard corpus. Gao et al. [10] analyzed characteristics of the longest location entity and identified it by using the maximum entropy model.

Above researches on Chinese place names recognition are in the standard news corpus, the characteristics of place names in the news is clear, easy to recognize, such as "Beijing" and "city of Zhengzhou in Henan province". Research on the text of complaint about urban management is markedly different from the research above. In addition to diverse format, location entity is complicated and longer. Identifying the location entity is difficult by using traditional methods. In this paper, we divide location entity recognition into basic place name recognition and indicator lexicon construction. Then attachment connection algorithm was put forward, which is ACA, to connect basic place name with indicator and identify location entity.

3 Location Entity Recognition

As discussed before, we use the model of location entity recognition to identify location entity. The definition of the model is as follows.

3.1 The Model of Location Entity Recognition

For the convenience of description, we define following concepts:

Definition 1. Basic place name: it is generalized location where event occurs and its length is usually short, denoted BasePla, the set is BasePlaSet.

Definition 2. Indicator: It often appears after the BasePla. Meanwhile, its appearance can make the location where event occurs more exactly, but that it appears alone is meaningless, denoted IndicateLoc. The set denoted that IndicateLocSet. Suppose $\text{IndicateLocSet} = \{a_1, a_2, \dots, a_m, d_1, d_2, \dots, d_n, s_1, s_2, \dots, s_k\}$ then $\text{IndicateLoc} \in \text{IndicateLocSet}$. $\text{AreaSet} = \{a_1, a_2, \dots, a_m\}$, $\text{DirectionSet} = \{d_1, d_2, \dots, d_n\}$, $\text{SpotSet} = \{s_1, s_2, \dots, s_k\}$, $\text{AreaSet} \cap \text{DirectionSet} = \emptyset$, $\text{DirectionSet} \cap \text{SpotSet} = \emptyset$, $\text{SpotSet} \cap \text{AreaSet} = \emptyset$.

Where AreaSet is the set of area indicators containing some words that indicate specific range where event occurs; DirectionSet is the set of direction indicators containing some words that indicate specific direction where event occurs; SpotSet is the set of place indicators containing some words that indicate specific spot where event occurs.

Definition 3. Location Entity: It is a specific location where event occurs, denoted LocEntity. Which is defined as below.

$$\text{LocEntity} = \text{BasePlaSet} + \text{NormalWordSet} + \text{IndicateLocSet}$$

Where NormalWordSet is the set of NormalWord except for BasePla and indicators. Namely, $\text{NormalWord} \notin \text{BasePlaSet} \cup \text{IndicateLocSet}$. As defined before, $|\text{NormalWordSet}| \geq 0$ and $|\text{IndicateLocSet}| > 0$ ($|A|$ is the number of elements in set A). To sum up, LocEntity mainly includes the following three characteristics:

1. It is a noun or a noun phrase.
2. It is the longest description of a location.
3. It is associated with a specific event.

Definition 4. BI distance: It is the number of NormalWords that appear continuously between BasePla and IndicateLoc and near the IndicateLoc, denoted BI-Len.

For example, for the sentence:“马家堡西路角门西 地铁站 外面 东北角 的 丁字路口 的问题”(The problem of the northeast T-junction outside the Majiabu Jiaomen west subway), where $\text{LocEntity}=[\text{马家堡西路角门西地铁站外面东北角的丁字路口}]$ (the northeast T-junction outside the Majiabu Jiaomen west subway), $\text{BasePlaSet}=\{\text{马家堡西路角门西(Majiabu Jiaomen West)}\}$, $\text{IndicateLocSet}=\{\text{地铁站(Subway), 外面(Outside), 东北角(Northeast corner), 丁字路口(T-junction)}\}$, where $\text{SpotSet}=\{\text{地铁站(Subway), 丁字路口(T-junction)}\}$, $\text{AreaSet}=\{\text{外面}\}$ (Outside), $\text{DirectionSet}=\{\text{东北角}\}$ (Northeast corner), $\text{NormalWordSet}=\{\text{的}\}$, $\text{BI}_1\text{-Len}=0$, $\text{BI}_2\text{-Len}=0$, $\text{BI}_3\text{-Len}=0$, and $\text{BI}_4\text{-Len}=1$.

In the next subsections, we will apply the divide-and-conquer strategy to build our model. The following three steps must be taken:

1. Identifying BasePla based on CRF role labeling;
4. Building indicator lexicon semi-automatically;
5. Identifying LocEntity using attachment connection algorithm.

3.2 BasePla Recognition Based on CRF Role Labeling

Since the BasePla recognition can be converted into sequence annotation and the boundary identification. Similarly, CRF is a kind of conditional probability model for annotation and segmentation ordinal data, which combines the characteristics of the maximum entropy model with hidden markov model, joins the long-distance contextual information, and solves the problem of label bias. So we use CRF model to identify BasePla.

The basic idea of BasePla recognition based on CRF role labeling is as follows: firstly, we process the corpus with Chinese word segmentation and part-of-speech tagging; secondly, some words are labeled with some roles using restrained role labeling; finally, we select word, part-of-speech as features to identify BasePla based on CRF.

The Definition of BasePla Roles. Roles table is the basis of the BasePla recognition. The computer can understand the message hidden in the Chinese characteristics according to roles table. Roles table mainly contains following roles:

1. Internal information of BasePla: A part of BasePla, namely tail word of BasePla, marked by W. For example, “红莲北里”(Red lotus North), “海淀区莲花小区”(Haidian District, Lotus area) and “大望路”(Da Wang road) .
- 2 External information of BasePla: Some words except for BasePla, including context , indicator and conjunctions.
 - (a) Context : Some words before and after the BasePla, tagged by SL and XR respectively, such as “至车碾店胡同”(To Cheniandian Lane) and “北京市平谷区供暖”.(Beijing Pinggu District Heating)
 - (b) Indicator : It usually appears behind the BasePla. On the other hand, it is divided into three kinds: area indicator, direction indicator and spot indicator, respectively, with QI, FI, DI annotations. such as “马家堡西路角门西地铁站外面东北角的丁字路口”(the northeast T-junction outside the Majiabu Jiaomen west subway)
 - (c) Conjunction: It refers to some words connecting two parallel place names, such as “和(and)、与(and), 及(and), 或(or), 或者(or), ‘、’, ……”, marked with C, for example, “帝京路和宝隆路”(Teikyo road and Baolong Road) .
- 3 Part-of-speech(POS) information of BasePla: The word that POS is ns marked with S. such as “海淀/ns五路居”(Haidian/ns Fifth Avenue Home).
- 4 Words not related to role: It is not related to the role, marked by N.

To conclude, in this paper we define 9 roles, as shown in table 2.

Restrained Role Labeling. Definition 5. Restrained role labeling: A word is conditional marked with a specific role, rather than any conditions.

Definition 6. Bag of words: the set of unrepeated words that extracted from the texts before and after a word as the specific window, denoted that Bag-w.

Table 2. Roles table

Role	Description	Example
W	Tail word	红莲 <u>北里</u> (Honglian North), <u>大望路</u> (Dawang road)
QI	Area indicator	朝阳 <u>十里堡地区</u> (Shilipu Chaoyang district), 戎晖家园 <u>周边</u> (surrounding of Rong Hui Home)
FI	Direction indicator	朝阳路 <u>北侧</u> (the north side of Chaoyang Road), 京洲北街 <u>南侧</u> 人行道(On the south side of pavement of Jingzhou North Street)
DI	Spot indicator	西红门宜家 <u>工地</u> (site of IKEA in the West Red Door), <u>定福庄路土路</u> (dirt road of Dingfuzhuang road)
SL	Words before the BasePla	<u>位于方庄东路</u> (Located Fangzhuang East), <u>家住房山区长阳镇</u> (one who lives in the town of Changyang in the Fang Shan area)
XR	Words after the BasePla	<u>西大望路交口处</u> (the intersection of West Dawang Road), <u>长安街邻近区域</u> (near Chang'an Avenue)
C	Conjunction	<u>帝京路</u> <u>和</u> <u>宝隆路</u> (Teikyo road and Baolong Road)
S	BasePla	<u>海淀/ns</u> 五路居(Haidian/ns Fifth Avenue Home)
N	Words not related to roles	

By observing the corpus, we find some POS like nr and nz are incorrect and they should be labelled nz, therefore we first constraint the POS role, the rules are as following:

1. POS constraint

- (a) Words with nr POS: nr is the POS of a person name, but person name seldom appears in the text of complaint about urban management. The word that POS is nr is mostly place or component of place. For example, “马连道/nr 中里”(Ma Liandao/nr Zhong), “马/nr 家堡西路”(Ma/nr Jiabao West). So we should convert nr into ns.
- (b) Words with nz POS: nz is the POS of institution names, but sometimes institution names can be indicated BasePla, for instance, the sentence “北京信息科技大学/nz 有多少学生”(How many students are Beijing information science and technology university), where “北京信息科技大学”(Beijing information

science and technology university) is institution name, but for the sentence “北京信息科技大学/nz 向东200米”(Beijing information science and technology university eastbound 200 meters), where “北京信息科技大学”(Beijing information science and technology university) is BasePla, so we need to convert nz POS into ns by using rule below: For any word which POS is nz, we obtain the Bag-w by setting the window to 4. If Bag-w contains IndicateLoc (the construction of indicators is shown 3.3), we convert nz into ns; or we do nothing.

As described before, role is related to the BasePla and appears near the BasePla, that is to say, role appears in the Bag-w of BasePla. Moreover, the words in the Bag-w of BasePla hold the 65% proportion in all words. If they are all marked with corresponding roles, the feature will not be clear between role words and normal words and the result is bad. We need to choose useful words according to following constrains.

2 Tail word and context constraint

We apply the formula 1 to statistic the probability of a word which is the tail word and get tail words table.

$$P = \frac{TF(\text{tail})}{TF(\text{all})} \quad (1)$$

Where TF(tail) is the count of a word which is the tail word, TF(all) is the total count of a word. The statistics of probability of a word that is the word before or after BasePla is the same as formula 1.

3 Conjunctions constraint

Since not all conjunctions connect two BasePla, we follow the regulation below to get useful conjunction: for all the conjunction, we set the window to 4 and get the set of POS, denoted PosSet. If PosSet contains the POS of ns, the conjunction is useful, and marked with C, or it is marked with N.

Since a word usually plays more than 2 roles, which will result in the difficulty in identifying BasePla. So we define that a word only plays the most important role which it can and sort the different roles according to the importance: tail word > indicator > conjunction > context word.

Feature Template of CRF. As mentioned before, CRF can connect the long-distance contextual information and combine various information, related or unrelated, therefore, we select word, POS and role as features, use B, I, E, O as label, where B is the begining of BasePla, I is the middle of BasePla, that is, between the begin and end, E is the tail of BasePla, O is the word that is not related to BasePla, and apply atom feature template and compound feature template, as shown in table 3, to identify BasePla using CRF.

Table 3. Feature template of CRF

Atom feature template	W(i), W(i+1), W(i+2), W(i+3), W(i-1), W(i-2), W(i-3), P(i), P(i+1), P(i-1), R(i), R(i+1), R(i+2), R(i-1), R(i-2)
Compound feature template	W(i-1)+P(i-1), P(i-1)+P(i), P(i)+P(i+1), R(i-2)+R(i-1)+W(i), W(i)+R(i+1)+R(i+2)

Where W is word, P is POS and R is role. W(i) is current word, W(i+1) is the first word after the current word and W(i-1) is the first word before the current word. R and P is the same to W.

3.3 Extraction and Expansion of Indicators

Indicator describes specific location where event occurs, we category functionally three types: area indicator, direction indicator and spot indicator. In this paper we extract indicators using semi-automatic method and expand indicator using Synonymy Thesaurus.

Extraction of Indicators. Indicator usually appears after BasePla and is limited to 3 types. In this paper, we extract five words after every BasePla to build Bag-w of indicators. However, not all words in Bag-w of indicators are indicators. After that, we would select some words containing some special characters as indicator. As mentioned before, different from NormalWord, indicator usually contains some special characters, for instance, area indicator usually contains “区”(area), “内”(inside) and “外”(outside);direction indicator usually contains “东”(east), “西”(west), “南”(south), “北”(north), “上”(up), and “下”(down); spot indicator is usually noun or noun phrase. In this paper, we obtain right indicators using features above and proofreading manually.

Expansion of Indicators. Since the number of indicators extracted from limited corpus is small, we need to expand indicators using external resources. In this paper we apply HIT-CIR Tongyici Cilin (Extended) to find synonymous and similar words and expand indicators.

Synonymy Thesaurus was written by Mei[11] in 1983, then HIT-CIR finished HIT-CIR Tongyici Cilin (Extended) based on which by introducing other external resources. The format of HIT IR-Lab Tongyici Cilin (Extended) is shown in table 4.

Table 4. The format of HIT IR-Lab Tongyici Cilin (Extended)

Codes	Similar words
Cb01D01@	时空(time)
Cb02A02=	东(East) 东边(East) 东方(East) 东面(East) 东头(East) 正东(East) 左(Left) 右上方(the upper right hand corner) 右上角(the upper right hand corner)
Bc02C17#	右下方(the lower right hand corner) 左上方(the upper left hand corner) 左上角(the upper left hand corner) 左下方(the lower left hand corner) 左下角(the lower left hand corner)

The method of expansion of indicators is as follows: For every character or word in the indicators, if HIT IR-Lab Tongyici Cilin (Extended) contains, we add its synonymous or similar words to indicators; or we do nothing.

As mentioned before, LocEntity consists of BasePla and indicator. Meanwhile, we have obtained BasePla and indicator as described before. In the next subsections, we will identify LocEntity by connecting BasePla with indicator.

3.4 LocEntity Recognition

Definition 7. Attachment relationship: For wordA and wordB, wordA is a meaningful word and it can appear alone, however, wordB is meaningless if it appears alone and it must attach itself to wordA. That is to say, wordB is a supplement to wordA and it makes wordA more specific. In brief, wordB depends on wordA, denoted wordB \rightarrow wordA. For example, “朝阳路北侧”(Chaoyang road north), where “北侧(north)” is a supplement to “朝阳路(Chaoyang road)” and “北侧(north)” is meaningless when it appears alone, that is, “北侧(north)” attaches itself to “朝阳路(Chaoyang road)” to make the place more specific, denoted that 北侧(north) \rightarrow 朝阳路(Chaoyang road).

Indicator is a supplement to BasePla in the corpus of complaint about urban management. On the other hand, indicator attached itself to BasePla, denoted indicator \rightarrow BasePla. So we propose Attachment Connection Algorithm, namely ACA, to connect BasePla with indicator.

ACA is as follows:

Algorithm 1. Attachment Connection Algorithm.

```

1: Input: every sentence Sen =  $W_1W_2W_3...W_n$  in the test corpus
TestC, BasePla =  $W_i...W_j$  ( $1 \leq i \leq j \leq n$ ), IndicateWSet =
{Indicate $W_1$ , Indicate $W_2$ , ..., Indicate $W_n$ }
2: BI-Len  $\leftarrow$  0, the position of connection pointer  $\leftarrow$  0,
LocEntity  $\leftarrow$  BasePla
3: for m  $\leftarrow$  j+1 to n do
4:   for m < n or BI-Len  $\leq$  4 do
5:     if  $W_m \in$  IndicateWSet then
6:       BI-Len  $\leftarrow$  0
7:       pointer  $\leftarrow$  m
8:     else BI-Len++
9:   endif
10: m++
11: LocEntity  $\leftarrow$  LocEntity+Wj+1...Wpointer
12: Output: LocEntity

```

Where the reason why BI-Len is less than or equal to 4 is the number of continuous NormalWords between BasePla and indicator is limited to 4 by analyzing LocEntity.

Long LocEntity may contain punctuation and it will be divided into two short LocEntity by punctuation when identifying LocEntity, for example “崇文区忠实里东区（东环居苑）小区门口路两侧”(the sides of the door of Chongwen District in the area of Dongshili east(East loop Home)) will be divided into “崇文区忠实里东区”(Chongwen District in the area of Dongshili east) and “东环居苑”(East loop Home).

In order to recall long LocEntity described above, we combine short LocEntity in accordance with the following rules:

1. There only exists a punctuation between two short LocEntity.
2. Short LocEntity co-exists in a sentence.

4 Experimental Results and Analysis

4.1 Preparation for Experiments

The experimental corpus is 1500 texts of complaint which are crawled from government web site. At the same time, two annotations are made: first label BasePla, second label LocEntity, and final two annotations are proofread by professional staff.

We select randomly 1000 texts of complaint as training data and the rest 500 texts as test data. Then, using NLPiR[12], that is ICTCLAS2013, to process Chinese corpus with Chinese word segmentation and part-of-speech tagging with PKU second standard. After that, we use role labeling to label the corpus. After repeated experiments, we set the value of tail constraint and context constraint to 0.4 and 0.5 respectively.

4.2 Evaluation Criterion

In this paper we use precision rate, recall rate and F-value to evaluate experimental result. (For brief, we use the P, R and F instead hereafter respectively)

$$P = \frac{NR}{NG} \times 100\% \quad (2)$$

$$R = \frac{NR}{NC} \times 100\% \quad (3)$$

$$F = \frac{2 \times P \times R}{P + R} \times 100\% \quad (4)$$

Where NR is the number of right LocEntity identified by our method, NG is the number of LocEntity identified by our method, NC is the number of LocEntity contained in the corpus.

4.3 Results and Analysis

CRF is a fusion of the characteristics of maximum entropy model and hidden markov model[13]. It combines contextual information with the advantages of maximum entropy model. So we choose CRF-based LocEntity recognition as contrast test, whose features and template are the same as CRF-based BasePla recognition.

The experimental results of BasePla and LocEntity are shown in table 5.

Table 5. Experimental results of BasePla and LocEntity

Experiment	R	P	F	
CRF	BasePla Recognition	85.35%	75.80%	80.29%
	LocEntity Recognition	85.04%	75.52%	80.00%
LocEntity Recognition using our method	88.77%	81.15%	84.79%	

As seen in the above table, the result indicates that our method is far better than CRF. Moreover, it gains up to 84.79% in F-value and 4.79 % higher than CRF-based. CRF-based LocEntity in F-value is 0.29% lower than CRF-based BasePla. The reason are as follows:

1. The feature of BasePla is clearer than LocEntity.
2. It is easy to identify by CRF for the entities which features appears in training data. For example “宣武区鸭子桥南里小区” (area of Duck Bridge South in Xuanwu District), “朝阳区都城心屿小区西侧停车场” (parking lot in the west of Chaoyang District Ducheng Xinyu area), “马家堡西路角门西地铁站外面的丁字路口”(T-junction west Majiabu Jiaomen outside the subway station). It is difficult to identify by CRF for the entities which features don't appear in training data, for instance, “马家堡西路角门西地铁站外面的丁字路口，南北向人行横道上北面”(T-junction outside the Majiabu Jiaomen West subway, in the north of North-south pedestrian crossing) and “西城区百万庄南街3号楼最东面”(On the extreme east 3rd floor of South Street in Xicheng District Baiwanzhuang).
- 3 By introducing ACA, our model can identify LocEntity, like “马家堡西路角门西地铁站外面的丁字路口，南北向人行横道上北面” (T-junction outside the subway station in the Majiabu Jiaomen West, in the north of North-south pedestrian crossing) and “西城区百万庄南街3号3楼最东面” (On the extreme east 3rd floor of South Street in Xicheng District Baiwanzhuang). This improves recall rate.

Table 6 shows some LocEntity extracted by our method.

Table 6. Identified LocEntity

Number	Identified LocEntity
1	定慧福里小区南门的停车场(Parking lot of South Gate in Dinghuifuli Distinct)
2	朝阳区劲松二区229号楼都城心屿小区西侧停车场(parking lot in the west of Chaoyang District Ducheng Xinyu area)
3	北京邮电大学南门对面胡同(the alley across the south gate of Beijing University of Posts and Telecommunications)
4	海淀区车道沟桥，牛顿办公区和嘉豪国际中心的停车场(Parking lot of Chedaogou Bridge, Newton office and Jiahao International Center)
5	马家堡西路角门西地铁站外面的丁字路口，南北向人行横道上北面 (T-junction outside the subway station in the Majiabu Jiaomen West, in the north of North-south pedestrian crossing)
6	西城区百万庄南街3号楼最东面(On the extreme east 3rd floor of South Street in Xicheng District Baiwanzhuang)

As can be seen from table 6, we can find that

1. Our approach can identify LocEntity with punctuation, such as “海淀区车道沟桥，牛顿办公区和嘉豪国际中心的停车场” (Parking lot of Chedaogou Bridge, Newton office and Jiahao International Center) and “马家堡西路角门西地铁站外面的丁字路口，南北向人行横道上北面” (T-junction outside Majiabou Jiaomen West subway, in the north of North-south pedestrian crossing).
- 3 Our approach can identify such LocEntity that CRF can't, such as “西城区百万庄南街3号楼最东面” (On the extreme east 3rd floor of South Street in Xicheng District Baiwanzhuang) and “海淀区车道沟桥，牛顿办公区和嘉豪国际中心的停车场” (Parking lot of Chedaogou Bridge, Newton office and Jiahao International Center).

In conclusion, our model solves the difficulty of identifying LocEntity to certain degree and injects new ideas for LocEntity recognition.

However, our model also extracts incorrect LocEntity such as “丰台区蒲黄榆二里这个小区”(this area of Huangyuerli in Fengtai District) and “朝阳区广顺北大街大西洋新城南门应该修建过街天桥”(The Atlantic Metro South Gate should be built overpass in Beijing chaoyang district wide BeiDaJie). That is because the people who complaint the problem of urban management using the word in the different ways and sometimes they use the modifiers before indicator of LocEntity, for example, the pronouns “这个”(this) and verb “修建”(build).

5 Conclusions and Future Work

In this paper, we propose a novel method to identify LocEntity by introducing indicators. Unlike traditional methods, our method follows the divide-and-conquer strategy: we divide LocEntity recognition into BasePla recognition and indicator lexicon construction. First, we use the CRF to identify BasePla, and then we build indicator lexicon semi-automatically. Finally, we propose the ACA to connect BasePla with indicator and obtain LocEntity.

Experiments on the corpus of complaint about urban management indicate that our method can not only ensure higher accuracy but also improve the recall rate. Moreover, the proposed method injects new ideas for LocEntity recognition. However, since our method depends on the precision of BasePla based on CRF and the comprehensive construction of indicators, we plan to expand training data and indicators so that the effect of recognition will be better in the future. Furthermore, the method is corpus independent and can be extended to other corpus once we have the training data in the target corpus.

Acknowledgement. This work is supported by National Natural Science Foundation of China under Grants No. 61271304, Beijing Natural Science Foundation of Class B Key Project under Grants No. KZ201311232037, and Innovative engineering of scientific research institutes in Beijing under Grants No. PXM2013_178215_000004.

References

- Cai, H.L., Liu, L., Li, H.: Rule Reasoning-based Occurring Place Recognition for Unexpected Event. *Journal of the China Society For Scientific and Technical Information* 30(2), 219–224 (2011)
- Li, L.S., Huang, D.G., Chen, C.R., et al.: Research on method of automatic recognition of Chinese place names based on support vector machines. *Minimicro Systems-Shenyang* 26(8), 1416 (2005)
- Tang, X.R., Chen, X.H., Xu, C., et al.: Discourse-Based Chinese Location Name Recognition. *Journal of Chinese Information Processing* 24(2), 24–32 (2010)
- Du, P., Liu, Y.: Recognition of Chinese place names based on ontology. *Xibei Shifan Daxue Xuebao/ Journal of Northwest Normal University (Natural Science)* 47(6), 87–93 (2011)
- Li, N., Zhang, Q.: Chinese place name identification with Chinese characters features. *Computer Engineering and Applications* 45(28), 230–232 (2009)
- Li, L.S., Dang, Y.Z., Liao, W.P., et al.: Recognition of Chinese location names based on CRF and rules. *Journal of Dalian University of Technology* 52(2), 285–289 (2012)
- Li, L.S., Huang, D.G., Chen, C.R., et al.: Identifying Chinese place names based on Support Vector Machines and rules. *Journal of Chinese Information Processing* 20(5), 51–57 (2006)
- Huang, D.G., Yue, G.L., Yang, Y.: Identification of Chinese place names based on statistics. *Journal of Chinese Information Processing* 17(2), 46–52 (2003)
- Qian, J., Zhang, Y.J., Zhang, T.: Research on Chinese Person Name and Location Name Recognition Based on Maximum Entropy Model. *Journal of Chinese Computer Systems* 27(9), 1761–1765 (2006)
- Gao, Y., Zhang, W., Zhang, Y., et al.: Application of Maximum Entropy Model in the LLE Identification. *Journal of Guangdong University of Petrochemical Technology* 4, 014 (2012)
- Mei, J.J., Zhu, Y.M., Gao, Y.Q., et al.: *Synonymy Thesaurus*. Shanghai Lexicographical Publishing House, Shanghai (1993)
- NLPIR Chinese word segmentation system, <http://ictclas.nlpir.org/downloads>
- Xu, B.: Oral standardization processing based on Conditional Random Fields model. Nanjing University of Science and Technology (2009)

Detect Missing Attributes for Entities in Knowledge Bases via Hierarchical Clustering^{*}

Bingfeng Luo^{1,**}, Huanquan Lu^{1,**},
Yigang Diao², Yansong Feng³, and Dongyan Zhao³

¹ Department of Machine Intelligence, Peking University, Peking, China
luoluo123n@pku.edu.cn, huanquanlu@g.ucla.edu

² Technic and Commnication Technology Bureau, Xinhua News Agency, China
thomasfred@xinhua.org

³ ICST, Peking University, Peking, China
{fengyansong,zhaody}@pku.edu.cn

Abstract. Automatically constructed knowledge bases often suffer from quality issues such as the lack of attributes for existing entities. Manually finding and filling missing attributes is time consuming and expensive since the volume of knowledge base is growing in an unforeseen speed. We, therefore, propose an automatic approach to suggest missing attributes for entities via hierarchical clustering based on the intuition that similar entities may share a similar group of attributes. We evaluate our method on a randomly sampled set of 20,000 entities from DBPedia. The experimental results show that our method can achieve a high precision and outperform existing methods.

Keywords: Missing Attributes, Automatic Knowledge Base Construction, Hierarchical Clustering.

1 Introduction

The proliferation of knowledge-sharing communities such as Wikipedia and the progress in scalable information extraction and machine learning techniques have enabled the automatic construction of knowledge bases (KBs) such as DBPedia, Freebase and YAGO[3]. However, these automatically constructed KBs often suffer from quality issues such as the lack of attributes, duplication of entities, incorrect classifications and et al. In this paper, we focus on the problem of lack of attributes for existing KB entities. Since the contents of KBs are often dynamic and their volume grows rapidly, manually detecting missing attributes is a labor-intensive and time consuming task. Therefore, it would be of great importance to automatically suggest possible missing attributes for entities in KBs.

^{*} This work was supported by the National High Technology R&D Program of China (Grant No. 2012AA011101, 2014AA015102), National Natural Science Foundation of China (Grant No. 61272344, 61202233, 61370055) and the joint project with IBM Research. Corresponding author: Yansong Feng.

^{**} Contributed equally to this work.

Once the missing attributes are found, the automatic KB construction systems can use these information as important hints to support the work of automatic information extraction. In other words, they can find the corresponding attribute values from the web or via other approaches for entities in the KB.

In this paper, we propose an effective way to find missing attributes for entities in the KB, and the experimental results show that our method can achieve a relatively high precision. We will show some related work in the following section and the description of our method will be demonstrated in the third section. After that, we will show the performance of our method with an experiment on DBPedia. Finally, in the last part, the conclusion of the paper will be made and some possible future work will be discussed.

2 Related Work

We define our task as follows: Given a KB and an entity described by a set of attribute-value pairs, the task is to detect appropriate missing attributes for this entity, which are currently not in its attribute set.

Most related work focuses on finding missing attribute values through heuristic ways[5] or just focuses on finding attribute-value pairs in the information extraction way[6]. Although the attribute plays an important part in both of these methods, there is very limited work that focuses directly on finding missing attributes. Recent work that focuses on this problem tries to suggest missing attribute candidates using association rule mining[1]. The basic idea is that if an entity has attribute A, B and etc, then it may imply that this entity should also have attribute C. For example, if an entity (a person in this case) has attribute *music genre* and attribute *instrument*, it may imply that he or she should also have the attribute *record label*. This can be captured by the association rule $musicgenre, instrument \rightarrow recordlabel$, which means that that if we have learned the association rule , then it is reasonable to assume that entities which have attribute *music genre* and attribute *instrument* should also attribute *record label*. Using the method described above, Abedjan and Naumann(2013)[1] obtain top 5 or 10 candidates for each entity. Since the top 5 or 10 candidates usually contains only a relatively low ratio of correct predictions, human examination is required after the candidates are generated.

In essence, the association rules make sense mainly because the left part of the association rule implies that the entity belongs to some categories, and most entities in these categories tend to have the attribute lies on the right part of the association rule. To be specific, having attribute *music genre* and *instrument* implies that this entity is probably a singer or a bandsman. Therefore, it is reasonable to infer that this entity should also have the attribute *record label*. However, this association rule method can only lead to a limited number of categories since it doesn't take advantage of attribute values. For instance, the attribute *occupation* says nothing but this entity is a person in this case. However, the values of attribute *occupation* can tell us if this person is a musician, a politician or something else. Another drawback of this method is that it

implicitly maps attribute set to categories, which will introduce some unnecessary errors to the predicting system. Therefore, we introduce a clustering-based method to overcome these drawbacks. In this way, we focus directly on the categories themselves, and we can base our results on a large number of categories.

3 Our Approach

Rather than inducing missing attributes via existing attributes as described in [1], we turn to use categories that an entity belongs to. Our solution is based on the intuition that if an attribute is owned by the majority of entities in a category, i.e., a group of similar entities, then this attribute should probably be owned by other entities in this category. For example, as shown in figure 1, we can see that *Taylor Swift* (a famous American singer) belongs to category *pop singer*, and almost all the entities except for *Taylor Swift* in category *pop singer* have the attribute *record_label*. Therefore, *Taylor Swift* probably should also have the attribute *record_label*. Notice that all the attributes shown in figure 1 and other examples afterwards come from the attribute system of Wikipedia

In our system, as shown in figure 2, we first convert each entity in the KB into continuous vector representations. Then, we apply a hierarchical clustering method to group entities into clusters according to different attributes, yielding clusters on the bottom of figure 1. Finally, in each cluster that the entity belongs to, we use the method described in figure 2 to detect missing attributes.

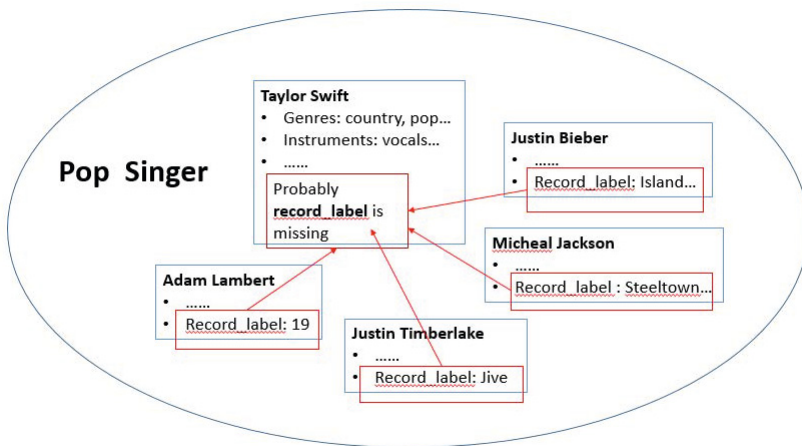


Fig. 1. An example of finding missing attributes for Taylor Swift

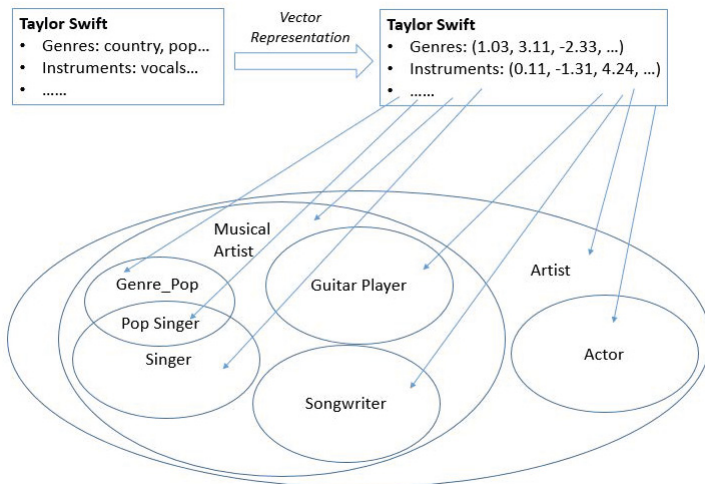


Fig. 2. Overview of our method

3.1 Preprocessing

Here we only make use of the attributes whose values are word strings when building cluster system. We learn a continuous vector representation (word embedding) for these values via RNNLM[2]. For example, the word string *actor* can be represented by a 200-dimension vector using RNNLM. After that, each entity is represented by a set of vectors, and each vector corresponds to a certain attribute of the entity. Notice that we simply abandon attributes with time and numerical values for the reason that they hardly yield useful clusters in this task. Actually, these values can be easily fitted into vector representation by using the first element to store the numeric value and setting other elements to be zero. As for time values, we can use the first element of a vector to represent year, the second to represent month, the third to represent day and the rest elements will be set to zero.

3.2 Entity Clustering

Why not Human-Made Category System: Since most existing KBs have their own category systems, using these existing category systems seems to be a plausible idea. However, these human made category systems have some disadvantages that make them unfit for our task.

First, the categories in these systems are not evenly distributed over all entities. These human made category systems tend to be very good in hot topics, but as for those less popular ones, these systems usually have very limited category information which is unlikely to grow in a short period of time. For example, the entity *Sinsharishkun*, an Assyrian King, has only 3 categories in Wikipedia

while the entity *Alexander the Great* has 16 categories. However, unfortunately, entities in these less popular topics are exactly the ones that are in strong need of missing attribute completion.

Second, the human-made category systems are not well-organized. Take Wikipedia again for example. The entity *Carson Henley*, an American, is assigned to the category *United States*. However, the category *United States* belongs to category *G20 nations*, which makes *Carson Henley* a nation. Actually, this kind of inconsistency can be seen everywhere in the Wikipedia category system, since many categories belong to a higher-level category that may lead to inconsistency like category *United States*.

Third, it is also hard to add new entities into these category systems automatically, which means that it is difficult to use these category systems in different KBs and for newly generated entities. On the one hand, the criterions that people used to make these categories are hard to be understood by machine. We can only use some contextual information to approximate the criterions, which is as difficult as building a new automatically-constructed category system since they all require transforming the contextual information into the way that can be understood by machine. On the other hand, people sometimes use the information that doesn't exist in both the attributes or the attribute values of the entity to create categories. Therefore, it is almost impossible to capture the criterions that people used when making these categories.

Considering all the disadvantages of human-made category systems, it is better to automatically build a category system on our own which is well organized, easy to update and possibly more expressive than human-made ones (we will discuss this later).

Clustering Algorithm: We use an overlapped hierarchical density-based clustering method (similar to [4]) to form our category system (since we are using clustering method, we use the term cluster system instead of category system afterwards). The method is an improved version of the traditional density-based clustering that it allows overlap as well as a hierarchical structure.

Algorithm 1. Overlapped Hierarchical Density-Based Clustering Algorithm.

Input:

- The set of entities, E_n ;
- A group of thresholds, T_k ;

Output:

- Grouping entities with the same attribute value into the same cluster, yielding first-layer clusters, $C_{1.m_1}$;
 - 1: Traditional density-based clustering on $C_{1.m_1}$, yielding clusters of the second layer, $C_{2.m_2}$;
 - 2: Assigning isolated points and non-core points to their neighbor clusters if close enough;
 - 3: Using mean vector as representative vector for $C_{2.m_2}$;
 - 4: Repeat step 1-3 to get higher layer clusters until no new cluster is generated;
-

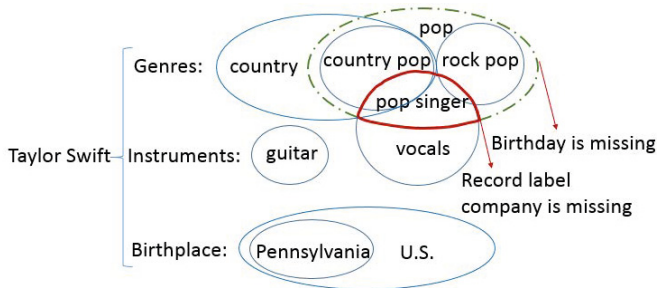


Fig. 3. An example for Taylor Swift’s attributes

Instead of considering all the attribute vectors when clustering entities, we deal with each individual attribute respectively. As shown in the algorithm above, for each attribute we group entities with the same attribute value into the same cluster (one entity may belong to several clusters), which forms the first layer of our hierarchical cluster system. Then, to construct the second layer, we first use traditional density-based method to get clusters without overlap. After that, we check if the isolated points as well as the non-core points are close enough to any clusters, and assign these points to their neighbor clusters which are close enough (one point can be assigned to several neighbor clusters). Now, the points that lies in the overlap areas are found. After that, we continue performing this overlapped density-based clustering method over these newly found clusters using their mean vectors as their representative vectors to get a higher layer. Repeat the former step until the algorithm doesn’t generate new clusters, then we get a hierarchy of clusters (*simple clusters*, to distinguish against *complex clusters* described afterwards) for each attribute, where each cluster contains entities that are similar to some extent. For example, figure 3 shows three attributes for *Taylor Swift*. According to her *genres*, she has been assigned to four clusters: *country*, *country pop*, *rock pop*, and *pop*. Note that *country pop* belongs to *country*, both *country pop* and *rock pop* belong to a higher-level cluster, *pop* and she is assigned to both *country pop* and *rock pop* for the reason that we introduced overlap into our algorithm.

Furthermore, notice that the intersection of different *simple clusters* is also meaningful. For example, the *pop singer* cluster is the intersection of *simple cluster pop* and *simple cluster vocals* in figure 3. Therefore, for each entity, we need to intersect combinations of its *simple clusters* to get *complex clusters* it may belong to. This intersecting step gives the cluster system great ability of expression that it can capture most of the categories in human-made category systems as well as those categories that people missed.

Obviously, it is inefficient to enumerate all combinations. We therefore apply a pruning strategy to improve the efficiency. The idea is that we can abandon small clusters when intersecting clusters. Because small clusters are not strong

enough to support our further induction. Empirically, intersecting five or more *simple clusters* rarely yields useful clusters, therefore, setting four as the maximum number of *simple clusters* to intersect is reasonable and is an effective way to speed up.

Good Properties of the Cluster System. First, our cluster system is well organized, which means there won't be any inconsistency described in the beginning of this section. This is guaranteed because we only cluster in the domain of one particular attribute once at a time, which makes us focus on just one aspect of the entities. On the contrary, human-made categories may focus on different aspects in different category level. In the example of section 3.2, the category system first focuses on nationality, then it changes to focus on international organization, which results in the inconsistency. In our cluster system, even if we will intersect these clusters afterwards, these new clusters still have clear and consistent focuses (previous focuses are always followed).

Second, our cluster system is possibly more expressive than human-made ones. This is achieved by applying intersection to *simple clusters*. On the one hand, as shown above, many human-made categories can be yielded by our algorithm. On the other hand, categories like *American Female Referee* (*complex clusters*) are only covered by human-made categories in a small proportion due to its huge amount. Therefore, our method can yield much more clusters than human-made ones. Furthermore, our method can easily yield clusters for less popular entities, which only have very limited number of human-made categories.

3.3 Detecting Missing Attributes

We define the attributes owned by the majority of entities of a cluster as *common attributes*. Once we have found appropriate cluster assignments for each entity, based on the intuition, we first compute the support of each attribute in a cluster, as follows:

$$\text{support}(a_i) = \frac{\text{number of entities that have } a_i}{\text{number of entities}}$$

If the support of attribute a_i is strong enough, then we consider a_i as a *common attribute* of this cluster, and a_i should be owned by all the members of the cluster. For example, since most pop singers have the attribute of *record label*, we can reasonably infer that all entities in cluster *pop singer* should have this attribute.

As for entities already existing in the KB, we have already found clusters they belong to during the clustering step. Then, for each entity, if a *common attribute* of the clusters that the entity belongs to does not appear in the entity's attribute set, we should suggest this attribute as its missing attribute.

For those newly extracted entities which have fewer attributes, we first find the *simple clusters* they may belong to by simply calculating the vector similarities, and find *complex clusters* by intersecting *simple clusters*. Notice that once it is

assigned to a lower-layer *simple cluster*, it should also be assigned to higher-layer *simple clusters* which encompass the lower-layer one. Once we have found all these clusters, we can take similar steps as those of existing entities to detect their missing attributes.

4 Experiments and Discussions

We examine our model on the *Person* category of DBPedia which contains randomly sampled 20,000 entities. We cluster half of entities in the data set, regarded as entities existing in the KB and the other half as new entities. We regard points that has more than 3 neighbor points with a similarity higher than 0.6 to be core points. When doing overlap, we lower the similarity threshold to 0.55. After that, we use the same parameter to get higher layer clusters. In the finding-missing-attribute part, we abandon clusters whose size is smaller than 5 and consider attributes with a support higher than 0.6 to be *common attributes*. When fitting new entities into the cluster system, we only accept the match with a similarity higher than 0.85.

To evaluate the performance of our models, we use the same evaluation strategy as [1]. We randomly drop one attribute from each entity in the data set, then we rank the suggested missing attribute candidates for each entity according to the support of these attributes within its clusters. If the top k candidates contain the dropped attribute, we record this instance as correct. We evaluate our method in different k s: top1, top5 and top10. Although the data set of [1] and ours are not identical, they are both randomly sampled from the *Person* category of DBPedia with similar volumes. We hope the imperfect comparisons could still draw the skeleton of differences between our method and the method of [1]. As Table 1 demonstrates, our method can achieve a precision of 84.43% by considering the first candidate, 95.36% for top 5 candidates and 96.05% for top 10 candidates for existing entities, much higher than [1] whose precision computed over the *Person* category is 51% for top 5 candidates and 71.4% for top 10 candidates([1] didn't compute precision for the first candidate). We can see that our method performs much better than [1], which makes sense because we focus directly on the categories themselves (rather than implicitly contained in the association rule), and we can base our results on a large number of categories.

However, for new entities, our method can only achieve a precision of 33.86% for top 1 attributes, 45.45% for top 5 and 46.07% for top 10, all of which are not reported by [1]. The experimental results show that the performance on

Table 1. The performance of our proposed model

Precision	Top 1	Top 5	Top 10	Human Evaluation
Association Rules	Not Available	51%	71.4%	Not Available
Existing Entities	84.43%	95.36%	96.05%	96.72%
New Entities	33.86%	45.45%	46.07%	97.09%

Table 2. Some missing attributes suggested by our model

Entity	the Most Important DBPedian Category	Missing Attributes
Cho Beom-Hyeon	Baseball Player	team_label, throwingSide, activeYearStartDate, deathDate, statisticValue
Joseph Kariyil	Christian Bishop	birthPlace
Gerry Joly	Singer-Songwriter	recordLabel_label, occupation_label, activeYearsStart Year, hometown_label
William Thum	Office Holder	successor_label, almaMater_label, residence_label
Bob Turner	Congressman	termPeriod_label, nationality_label, occupation_label, battle_label, allegiance
Andrew H. Ward	Congressman	nationality_label, residence_label, termPeriod_label, occupation_label, otherParty_label
Li Haiqiang	Soccer Player	height, weight, shoots, number, league_label
Julian L. Lapidés	Politician	almaMater_label, termPeriod_label
Tomo Yasuda	Musician	genre_label, occupation_label, recordLabel_label
Howard L. Vickery	Military Person	allegiance, occupation_label, militaryBranch_label, award_label

new entities decreases significantly. One of the reasons may be that the dropped attributes of new entities may never appear in the training data.

We should notice that, the evaluation criterion does not make full sense in the case when most of the attributes in top k are predicted correctly except for the imperfection that these k attributes don't contain the one we dropped before. And this situation occurs possibly because that the dropped attributes of

new entities may never appear in the training data. Therefore, we also perform human evaluation on the detected missing attributes.

In the human evaluation, we first randomly sample 2,000 entities, 1,000 from existing entities and 1,000 from the new ones. Then, we manually judge whether each suggested missing attribute is reasonable or not. As shown in table 2, we randomly selected 10 entities (the first 5 are existing entities and the last 5 are new entities) and chose the top 5 candidates for exhibition (some entities may have less than 5 candidates). We can see that most missing attributes are reasonable except for 3 imperfections. One is that we suggest *Bob Turner* should have the attribute *Battle*. This suggestion appears because he once served in the army for about 4 years. However, this service doesn't necessarily imply that he once attended a battle. Another one occurs in the results of *Andrew H. Ward*. The results contain the attribute *Other Party*, which apparently doesn't make sense here. The final one is associated with *Howard L. Vickery*, who has the missing attribute suggestion *Award*. This doesn't make full sense because not all military person receive an award. However, the result is right at this time.

In general, the accuracy for existing entities is 96.72% and the accuracy for new entities is 97.09%, which is high enough to support automatic knowledge base construction. Therefore, the first evaluation method actually underestimated the performance of our method. Apart from that, this human examination result to some extent supports our explanation for the relatively bad performance on new entities under the first evaluation criterion: the dropped attributes of new entities may never appear in the training data.

5 Conclusion and Future Work

In this paper, we propose to automatically suggest missing attributes for KB entities by examining *common attributes* owned by the majority of entities in their clusters obtained through a hierarchical clustering algorithm. The experimental results show that our model outperforms existing method by a large margin and human evaluation indicates the potentials of our model in practice.

However, we admit that the more challenging scenario for our model is to detect missing attributes in multi-language knowledge bases. Our next step is to adapt our ideas to Chinese data sets and investigate the possibility of applying to multi-language knowledge bases.

When missing attributes are found, we still need to fill the missing attribute values. Therefore, we will also try to combine our system with information extraction methods to find the missing attribute-value pairs for entities in the future.

Furthermore, our method actually forms the basis of solutions for other quality applications in the automatically constructed knowledge bases since the cluster system we built actually has more power than just detecting missing attributes. For instance, with our cluster system we can deal with errors in entity type classifications.

References

1. Abedjan, Z., Naumann, F.: Improving rdf data through association rule mining. *Datenbank-Spektrum* 13(2), 111–120 (2013)
2. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781 (2013)
3. Suchanek, F., Weikum, G.: Knowledge harvesting in the big-data era. In: *Proceedings of the 2013 International Conference on Management of Data*, pp. 933–938 (2013)
4. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.: Scan: A structural clustering algorithm for networks. In: *Proceedings of the 13th ACM SIGKDD*, pp. 824–833 (2007)
5. Grzymala-Busse, J.W., Grzymala-Busse, W.J.: Handling Missing Attribute Values. In: *Data Mining and Knowledge Discovery Handbook*, pp. 33–51 (2010)
6. Wong, Y.W., Widdows, D., Lokovic, T., Nigam, K.: Scalable Attribute-Value Extraction from Semi-Structured Text. *IEEE International Conference on Data Mining Workshops* (2009)

Improved Automatic Keyword Extraction Based on TextRank Using Domain Knowledge

Guangyi Li and Houfeng Wang

Key Laboratory of Computational Linguistics, Ministry of Education,
Institute of Computational Linguistics, School of Electronics Engineering and
Computer Science, Peking University
{liguangyi,wanghf}@pku.edu.cn

Abstract. Keyword extraction of scientific articles is beneficial for retrieving scientific articles of a certain topic and grasping the trend of academic development. For the task of keyword extraction for Chinese scientific articles, we adopt the framework of selecting keyword candidates by Document Frequency Accessor Variety (DF-AV) and running TextRank algorithm on a phrase network. To improve domain adaption of keyword extraction, we introduce known keywords of a certain domain as domain knowledge into this framework. Experimental results show that domain knowledge can improve performance of keyword extraction generally.

Keywords: Keyword Extraction, TextRank, Domain Knowledge.

1 Introduction

Keywords, consisting of one single word or several words, summarize topics and ideas of an article. Keywords can benefit many NLP applications, such as text categorization, document clustering, search engine, etc. In an era when information on Internet grows explosively, it is intractable to scan every document thoroughly. Keywords enable us to find documents we need from the ocean of information.

In order to capture the topics of an article accurately and sufficiently, keywords usually need to be assigned by experts with adequate domain knowledge. However, with innumerable documents emerging everyday, it would be too costly to assign keywords to documents by human efforts. Therefore, automatic keyword extraction is drawing interests of many researchers and a number of techniques are applied to this task successfully.

The targets of keyword extraction include news articles, web pages, scientific articles, etc. Study of keyword extraction for scientific articles is getting more attention recently, since keywords are essential for retrieving scientific articles and grasping the trend of academic development. Though keywords are usually required for scientific articles and academic dissertations, many authors have trouble selecting proper keywords. And different authors will give keywords following different criteria. An efficient keyword extraction system can aid authors

in selecting proper keywords and help to correct inadequate keywords given by authors.

For keyword extraction for scientific articles, a shared task was held at the Workshop on Semantic Evaluation 2010 (SemEval-2010 Task 5) [1]. It was geared towards scientific articles in English. However, research on keyword extraction towards scientific articles in Chinese is relatively rare. We extract keywords from Chinese scientific articles adopting the framework of selecting keyword candidates by Document Frequency Accessor Variety(DF-AV) and running TextRank algorithm on a phrase network. We show how to improve the result of unsupervised keyword extraction for a certain domain with domain knowledge of known keywords.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the general framework for keyword extraction. Section 4 shows why known keywords can be used as domain knowledge and how to improve the result of keyword extraction with domain knowledge. Section 5 represents our experiment results and Section 6 concludes our work.

2 Related Work

The task of keyword extraction is usually divided into two steps: candidate selection and keyword ranking. Most keywords are nouns or noun phrases. Therefore, for candidate selection, most work is based on n-gram [2–4] or Part-of-speech tags [5, 6] or both [7]. Especially, [8] compared n-gram, POS tags and NP-chunks and demonstrated that voting from the three methods performs best.

Choosing from keyword candidates is usually considered a ranking problem. Each candidate is assigned a score, and top-k ranked candidates are chosen as keywords. Statistics are commonly used feature for ranking, among which TF-IDF is the most popular feature [2, 6, 9, 10]. Word co-occurrence is another widely used feature [11–13]. Supervised learning methods are also adopted for keyword extraction, including maximum entropy [6], naive Bayes [2, 12], support vector machines [14], conditional random field [15], etc. [16] gives a summary of systems which participated in SemEval-2010 Task 5. The best performance is achieved by bagged decision tree [17].

TextRank [18] is a graph-based, unsupervised ranking algorithm. It performs well for keyword extraction and becomes popular recently. Related research includes [19], [20], etc.

3 General Framework

In this section, we describe the framework based on TextRank for keyword extraction for Chinese scientific articles. First, keyword candidates are selected from the document by Document Frequency Accessor Variety(DF-AV). Second, we build a phrase network using candidates and rank candidates with TextRank. Top-k ranked candidates are selected as extracted keywords.

3.1 Candidate Selection by DF-AV

Keywords of scientific articles are mostly noun phrases. As for English, defined POS sequences are used to select keyword candidates. However, for Chinese, this might not work well, as accuracy of POS tagging for Chinese scientific articles is not satisfactory. There are two main reasons. First, Chinese words have fewer morphological changes than English. For instance, verb "extract" and noun "extraction" will be translated to the same Chinese word. This brings difficulty to Chinese POS tagging. Second, most Chinese POS tagging systems are trained on news corpus, while many keywords of scientific articles rarely appear in news corpus, i.e., these words are Out-of-Vocabulary words. Therefore, POS tagging for Chinese words may contain many errors.

As a consequence, We use the statistical criterion instead of POS sequence to select keyword candidates. Accessor Variety(AV) [21] is a statistical criterion first used for new word extraction from Chinese text collections. The criterion is proposed from the viewpoint that a word is a distinguished linguistic entity which can be used in many environments. Therefore, the numbers of different characters appearing before and after a word is relatively high. Likewise, we can adopt Accessor Variety for keyword candidates selection.

For a phrase string phr , let S_L denote the set of words appearing before phr , S_R denote the set of words appearing after phr . Thus, left Accessor Variety of phr $AV_L = sizeof(S_L)$, and right Accessor Variety of phr $AV_R = sizeof(S_R)$. The larger Accessor Varieties are, the more likely phrase phr is a keyword candidate. We define score of phrase phr as $Score(phr) = Freq(phr) \times AV_L(phr) \times AV_R(phr)$. All phrases whose score is higher than a certain threshold are selected as keyword candidates.

However, criterion of Accessor Variety cannot deal with low-frequency phrases well, because it's easy to prove that all phrases appearing once in the document will get a score of one. As a consequence, it cannot distinguish proper keyword candidates from all low-frequency phrases. To solve this problem, we transform Accessor Variety(AV) into Document Frequency Accessor Variety(DF-AV).

We investigate how keywords are distributed across the document and discover that keywords are usually specialized words and words around keywords are usually common words. Document Frequency(DF) are usually used to distinguish specialized words and common words. It is generally admitted that words with high document frequency are usually common words. Therefore, if a phrase is surrounded by words with high document frequency, it's very likely to be a keyword candidate. This leads to DF-AV.

We calculate Document Frequency of words based on Chinese Gigaword corpus, which consists of around 1.4 million news articles. For a phrase string phr , define DF-AV and score of phr as follows:

$$DFAV_L = \sum_{w \in S_L} \log doc_freq(w)$$

$$DFAV_R = \sum_{w \in S_R} \log doc_freq(w)$$

$$Score(phr) = DFAV_L(phr) \times DFAV_R(phr)$$

A maximum length limits the number of words combining a phrase string. All phrases whose score higher than a threshold will be selected as keyword candidates. A low threshold will raise coverage rate of real keywords, but, on the other hand, it will result in more non-keyword phrases involved. Therefore, a proper threshold is needed to keep the balance.

3.2 TextRank on a Phrase Network

TextRank is a graph-based ranking algorithm inspired by famous PageRank [22]. TextRank transfers the document into a network of words, in which an edge between words stands for a relation between words. The importance of a word is determined by the importance of its neighbours. Formally, denote $G = (V, E)$ a directed graph with the set of vertices V and set of edges E . For a given vertex V_i , denote $In(V_i)$ the set of vertices with edges to V_i , $Out(V_i)$ the set of vertices with edges from V_i . The score of a vertex V_i is defined as follows:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where d is a damping factor between 0 and 1. TextRank can be computed either iteratively or algebraically, like PageRank. In previous work [18, 20], a vertex of the graph stands for a single word. keywords are generated from combinations of top-ranked words. This method cannot ensure all generated keywords are independent linguistic entities. Additionally, not all words within a keyword can be ranked among top k. To improve this, we run TextRank on a phrase network, ranking phrases directly.

Based on keyword candidates selected by DF-AV, we build a graph with vertices standing for phrases. Usually, co-occurrence of words within a window of n determines a link between the words. We extend this relationship to words and phrases. Take word sequence "A B C D E" as an example. Each letter stands for a word. Suppose "BC", "CD", "BCD" are keyword candidates selected by DF-AV. We build a neighboring graph as Fig.1 . A directed edge from one vertex to another vertex means the latter one is next to the former one on the sequence.

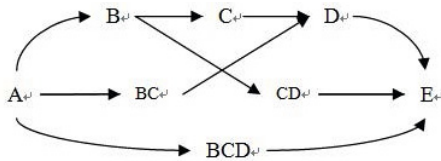


Fig. 1. Neighboring Graph

Based on neighboring graph, build phrase network graph according to window size n . To be specific, if there is a directed path no longer than n between two vertices, add a link between the vertices. Therefore, no linked vertices will share the same word. For instance, there are no links between "B", "BC", and "BCD". The phrase network graph based on Fig.1 with $n = 2$ is shown as Fig.2 .

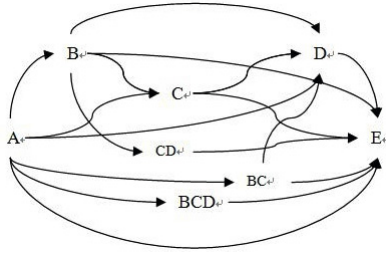


Fig. 2. Phrase Network Graph

Based on phrase network graph, we can compute importance of each vertex iteratively. Top-k ranked phrases are selected as keywords by the framework.

4 Improvement by Domain Knowledge

For scientific articles, choices of words vary between different domains. Especially, keywords are usually made up of specialized words, most of which are unique to the domain. Therefore, taking advantage of domain knowledge can improve performance of keyword extraction. In previous work [23, 24], thesaurus or Wikipedia are used as domain knowledge. They are usually of high quality but often not quite adaptive and construction of such resources is highly costly. However, through some online scientific article retrieval system, quantities of author-assigned keywords of a certain domain are available. Though some of those keywords are not quite normative, they can provide useful domain knowledge. In this section we'll show how to take advantage of raw known keywords to improve performance of keyword extraction.

4.1 Length of Keyword

There are many characteristics of keywords varying between domains. Length of keyword is a typical one. To show this characteristic, we choose four domains varying largely from each other, which are ethnology, petroleum, mathematics and international law. We do statistics of length of keywords from 1000 documents of each domain and show average length and distribution of length as Table.1 and Fig.3 respectively.

Table 1. Average Length of Keyword in Different Domains

Domain	ethnology	petroleum	mathematics	international law
Ave. Len.	3.54	4.16	4.62	4.48

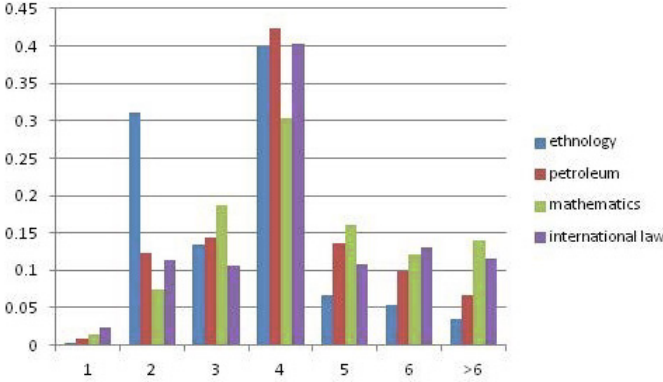


Fig. 3. Distribution of Length of Keyword

It's obvious length of keyword vary between domains. Keywords of ethnology tend to be short, while keywords of mathematics contain many longer ones. We will take advantage of this characteristic in two ways. First, we modify phrase-based TextRank graph into a weighted one. The new score is as follows.

$$WS(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_i)} w_{jk}} WS(V_j)$$

We define $w_{ij} = 0.5 + 0.5 \times n_{len(V_j)} / \max(n_k)$, where n_k is the number of keywords with length k for this domain. Second, we use the same weight as a multiplied factor to TextRank score. In this way, we can eliminate keywords that are too long or too short.

4.2 Components of Keyword

We discover that components of keyword also vary largely between domains. We do statistics on words forming keywords and find that distribution of words are unique to the domain. For example, the word "random" is the most frequent word appearing in keywords of mathematics, while it never appears in a keyword of ethnology. And the word "culture" is most likely to be seen in a keyword of ethnology, but it only appear twice in keywords of mathematics. What's more, in the same domain, words have different possibility to act as starting word or ending word of keywords. For example, "system" and "equation" are among most

frequent ending words of keywords, but they never act as starting word unless they act as keywords independently. These phenomena are apparently useful for keyword extraction.

To employ such information into keyword extraction, we use it to eliminate irrelevant keyword candidates. If a keyword candidate starts with or ends with a word that never appears in this position, or it contains a word that never appears in the domain, we will discard this candidate. In order that out-of-vocabulary words will not be discarded, we require that related words must be common words with high document frequency.

4.3 High-frequency Keyword

Some of the keywords are frequently selected as keywords, especially those words indicating area of research or popular method. Thesaurus of the domain is a common resource for such specialized words. However, not every domain has such a thesaurus and it's costly to build a thesaurus. At this time, we can take advantage of quantities of author-assigned keywords. statistics show that about half the keywords are selected as keyword more than once in a certain domain and the most frequent keyword serves about 1/20 of all documents.

Based on intuition that high-frequency keywords are more likely to be keywords of other documents, we increase TextRank score of such keywords. We multiply the score by a weight $w_f = \sqrt[3]{freq(phr)}$, where $freq(phr)$ is the frequency of phr . The top-k ranked keywords according to the weighted score are selected as the extracted keywords.

5 Experiments and Evaluations

In this section, we first introduce the experimental settings in detail. Then we present the experimental results and give an analysis.

5.1 Experimental Setting

Dataset There are a few datasets for keyword extraction in English. However, similar datasets for Chinese are rare. So we retrieve our data from *cnki.net*. We choose four domains, ethnology, petroleum, mathematics and international law. For each domain, we retrieve title, abstract and author-assigned keywords of 100 randomly-selected documents as test set and author-assigned keywords of another 1000 randomly selected documents as domain knowledge. It is notable that we discard documents whose keywords never appear in the abstract, since our method is an extraction method from text, which determines unseen keywords cannot be dealt with. We take author-assigned keywords as standard keywords, even though some of the keywords might be inappropriate.

Pre-processing. We used a perceptron-based tool implemented based on [25] to do word segmentation on all titles, abstracts and keywords. And we do statistics of known keywords to obtain domain knowledge. When calculating length of

keyword, we treat single English letter and punctuation as length 1 and a whole English word as length 2, since 2-character words are most frequent in Chinese.

Evaluation. Following evaluation method of SemEval-2010 Task 5 [1], we show P,R,F1 of top 5, top 10 and top 15 ranked keywords, where F1 is the harmonic average of P and R. When averaging F1 across documents, we calculate macro-average and micro-average of F1 and take the mean of macro-average and micro-average as metrics of performance.

5.2 Experimental Results

To demonstrate difficulty of keyword extraction of different domains, we adopt the method of Term Frequency Inverted Document Frequency(TF-IDF) as a baseline system. It ranks all phrase strings using TF-IDF and choose top-k ranked phrases as keywords. When extracting keywords using our framework. For candidate selection, we set maximum word number as 4 and threshold as 100 to keep a balance of recall and precision.

Based on Phrase-based TextRank, we add domain knowledge one by one to show its effects on keywords extraction. Our experimental results are shown on Table 2, in which + means add this information based on the above system.

Table 2. Experimental Results of Keyword Extraction in Different Domains

	ethnology			petroleum			mathematics			international law		
	Top5	Top10	Top15	Top5	Top10	Top15	Top5	Top10	Top15	Top5	Top10	Top15
TF-IDF	0.243	0.234	0.201	0.108	0.141	0.148	0.115	0.122	0.127	0.211	0.189	0.166
TextRank	0.312	0.249	0.201	0.179	0.184	0.173	0.167	0.176	0.173	0.287	0.238	0.197
+component	0.319	0.253	0.199	0.176	0.184	0.176	0.170	0.176	0.176	0.285	0.239	0.196
+length	0.326	0.256	0.203	0.181	0.186	0.176	0.172	0.179	0.178	0.290	0.242	0.197
+high-freq	0.342	0.258	0.205	0.202	0.201	0.180	0.180	0.187	0.183	0.300	0.249	0.199

The results show the framework based on TextRank over a graph network can extract keywords effectively. There is a major improvement over TF-IDF, though it seems that improvement of top 15 is relatively small, which is because average keyword number is around 5, leading to precision lower than 35% even for the best case. Domain knowledge simply from known keywords can improve performance of keyword extraction, and the improvement is especially significant for Top 5 results. It's a simple and effective way to improve keyword extraction result from an unsupervised method. However, as domain knowledge added one by one, improvement might not be so significant, because targets of different domain knowledge work on might be overlapped. Among three aspects of information, improvement of high-frequency keywords is obvious, while improvement of components is not very stable, because the number of known keywords is limited and it is impossible to cover every possible keyword composition characteristics.

Comparing between domains, we can see that performance of ethnology and international law is much better than the other two domains. tf-idf results show

directly that keywords are easily to extract from ethnology and international law via frequency method. We analyse this phenomenon and find some objective reasons. First, words from those two domains are more similar to news so that precision of word segmentation is better, while the other domains vary from news largely. Second, documents from petroleum and mathematics contain many English words and symbols, and document structure is more complicated. It adds difficulty to keyword extraction.

Though introducing domain knowledge shows improvement to keyword extraction, general performance of keyword extraction is not ideal, especially for petroleum and mathematics. How to narrow the gap between domains and improve performance is our next task. What's more, though our proposed way to take advantage of domain knowledge is simple and effective, it relies on coverage and quality of known keywords. We will investigate how to combine unsupervised and supervised methods to build a better keyword extraction system.

6 Conclusion

This paper shows how to improve TextRank based framework for keyword extraction on Chinese scientific articles using domain knowledg. We first select keyword candidates by DF-AV. Then, based on selected candidates, we build a phrase network graph and run TextRank algorithm to select top-k ranked keywords. We use known keywords as domain knowledge to improve keyword extraction, with information of length of keyword, components of keywords and high-frequency keywords. Experimental results show that domain knowledge can improve performance of keyword extraction generally.

Acknowledgments. This research was partly supported by National Natural Science Foundation of China (No. 61370117, 61333018), Major National Social Science Fund of China (No. 12&ZD227), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101).

References

1. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 21–26. Association for Computational Linguistics (2010)
2. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction (1999)
3. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 33–40. Association for Computational Linguistics (2003)
4. Paukkeri, M.S., Nieminen, I.T., Pöllä, M., Honkela, T.: A language-independent approach to keyphrase extraction and evaluation. In: COLING (Posters), pp. 83–86 (2008)

5. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Hamilton, H.J. (ed.) *Canadian AI 2000. LNCS (LNAI)*, vol. 1822, pp. 40–52. Springer, Heidelberg (2000)
6. Nguyen, T.D., Kan, M.-Y.: Keyphrase extraction in scientific publications. In: Goh, D.H.-L., Cao, T.H., Sølvsberg, I.T., Rasmussen, E. (eds.) *ICADL 2007. LNCS*, vol. 4822, pp. 317–326. Springer, Heidelberg (2007)
7. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 216–223. Association for Computational Linguistics (2003)
8. Hulth, A.: Combining machine learning and natural language processing for automatic keyword extraction. Department of Computer and Systems Sciences (Institutionen för Data-och systemvetenskap), Univ. (2004)
9. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: *Proceedings of the Fourth ACM Conference on Digital Libraries*, pp. 254–255. ACM (1999)
10. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: *Proceedings of human language technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 620–628. Association for Computational Linguistics (2009)
11. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools* 13(01), 157–169 (2004)
12. Ercan, G.: Automated text summarization and keyphrase extraction. PhD thesis, bilkent university (2006)
13. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 1, pp. 257–266. Association for Computational Linguistics (2009)
14. Krapivin, M., Autayeu, M., Marchese, M., Blanzieri, E., Segata, N.: Improving machine learning approaches for keyphrases extraction from scientific documents with natural language knowledge. In: *Proceedings of the Joint JCDL/ICADL International Digital Libraries Conference*, pp. 102–111 (2010)
15. Zhang, C.: Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems* 4(3), 1169–1180 (2008)
16. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation* 47(3), 723–742 (2013)
17. Lopez, P., Romary, L.: Humb: Automatic key term extraction from scientific articles in grobid. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 248–251. Association for Computational Linguistics (2010)
18. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. Association for Computational Linguistics (2004)
19. Wan, X., Xiao, J.: Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, pp. 969–976. Association for Computational Linguistics (2008)
20. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 366–376. Association for Computational Linguistics (2010)

21. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor variety criteria for chinese word extraction. *Computational Linguistics* 30(1), 75–93 (2004)
22. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
23. Hulth, A., Karlgren, J., Jonsson, A., Boström, H., Asker, L.: Automatic keyword extraction using domain knowledge. In: Gelbukh, A. (ed.) *CICLing 2001*. LNCS, vol. 2004, pp. 472–482. Springer, Heidelberg (2001)
24. Coursey, K.H., Mihalcea, R., Moen, W.E.: Automatic keyword extraction for learning object repositories. *Proceedings of the American Society for Information Science and Technology* 45(1), 1–10 (2008)
25. Zhang, Y., Clark, S.: Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics* 37(1), 105–151 (2011)

A Topic-Based Reordering Model for Statistical Machine Translation

Xing Wang, Deyi Xiong*, Min Zhang, Yu Hong, and Jianmin Yao

Provincial Key Laboratory for Computer Information Processing Technology
Soochow University, Suzhou, China
xingwsuda@gmail.com,
{dyxiong,minzhang,yhong,jyao}@suda.edu.cn

Abstract. Reordering models are one of essential components of statistical machine translation. In this paper, we propose a topic-based reordering model to predict orders for neighboring blocks by capturing topic-sensitive reordering patterns. We automatically learn reordering examples from bilingual training data, which are associated with document-level and word-level topic information induced by LDA topic model. These learned reordering examples are used as evidences to train a topic-based reordering model that is built on a maximum entropy (MaxEnt) classifier. We conduct large-scale experiments to validate the effectiveness of the proposed topic-based reordering model on the NIST Chinese-to-English translation task. Experimental results show that our topic-based reordering model achieves significant performance improvement over the conventional reordering model using only lexical information.

Keywords: statistical machine translation, reordering model, topic information.

1 Introduction

In recent years, phrase-based SMT has been widely used. It segments a source sentence into a phrase sequence, then translates and reorders these phrases in the target. Phrase reordering in the target is critical issue for SMT [6]. A great variety of models have been proposed to address this issue. Lexical, syntactical and semantic information are explored to predicate phrase orientations. Unfortunately these models only focus on sentence-level information and neglect document-level information.

Document-level information has proved very useful in many NLP tasks. In SMT literature, some researchers utilize document level information to improve translation and language models [4,5,13,15]. The translation knowledge which appears in the preceding sentences' translations is used to guide translations of

* Corresponding author.

the current sentence as well as the succeeding sentences. With regard to reordering models, document-level information has also been studied but in an implicit fashion. [12] propose a topic-based similarity model for translation rule selection in hierarchical phrase-based SMT. They report that topic information is not only helpful for phrase and monotone rule selection but also for reordering rule selection. [2] also find that phrase reorderings are sensitive to domains where they occur and that reordering model adaptation can improve translation quality. These two studies suggest that document-level information can be used for phrase reordering. However, to the best of our knowledge, there is no attempt to directly incorporate document-level information into reordering models. In this paper, we investigate how document-level information, especially topic information can be explicitly integrated into reordering models.

We therefore propose a novel topic-based reordering model to improve phrase reordering for SMT by using document-level topic information. We employ two kinds of topic information in our topic-based reordering model: (1) document topic assignment, and (2) word topic assignment. We integrate our topic-based reordering model into a state-of-the-art phrase-based SMT system. We train the system with large-scale Chinese-English bilingual data. Experimental results on the NIST benchmark data show that our topic-based reordering model is able to achieve substantial performance improvement over the baseline. We also find that both document topic features and word topic features are able to improve phrase reordering and the combination of these two groups of features obtains the best performance.

The rest of the paper is organized as follows. In section 2, we present the topic-based reordering model with novel topic features and detail the training process of the model. The integration of the proposed model into SMT system is presented in the section 3. We describe our experiments, including the details on experimental setup and present the experimental results in section 4. In section 5, we provide a brief analysis on the results. Finally, we conclude the paper in section 6.

2 Topic-Based Reordering Model

In this section, we present the proposed topic-based reordering model, different features that are used in our topic-based reordering model as well as the training procedure of the model.

2.1 Reordering Model

We discuss our topic-based reordering model under the ITG constraints [11]. The following three Bracketing Transduction Grammar (BTG) rules are used to constrain translation and reordering.

$$A \rightarrow [A^1, A^2] \quad (1)$$

$$A \rightarrow \langle A^1, A^2 \rangle \quad (2)$$

$$A \rightarrow f/e \quad (3)$$

where A is a block which consists of a pair of source and target strings. A^1 and A^2 are two consecutive blocks. The rule (1) and rule (2) are reordering rules which are used to merge two blocks into a larger block in a straight or inverted order. Rule (3) is a lexical rule that translates a source phrase f into a target phrase e and correspondingly generates a block A . A maximum entropy (MaxEnt) reordering model is proposed by [14] to predict orders for neighboring blocks under the ITG constraints. Following them, we also use a maximum entropy classifier to predict ITG orientation $o \in \{\textit{monotonous}, \textit{inverted}\}$ for two neighboring blocks A^1 and A^2 as MaxEnt is able to handle arbitrary and overlapping features from different knowledge sources. The classifier can be formulated as follows:

$$p(o|C(A^1, A^2)) = \frac{\exp(\sum_i \theta_i f_i(o, C(A^1, A^2)))}{\sum_{o'} \exp(\sum_i \theta_i f_i(o', C(A^1, A^2)))} \quad (4)$$

where $C(A^1, A^2)$ indicates the attributes of A^1 and A^2 , f_i are binary features, θ_i are weights of these features.

2.2 Features

We integrate three kinds of features into the topic-based reordering model: 1) boundary word features; 2) document topic features (DT) and 3) word topic features (WT). All these features are in the following binary form:

$$f_i(o, C(A^1, A^2)) = \begin{cases} 1, & \textit{if } (o, C(A^1, A^2)) \textit{ satisfies certain condition} \\ 0, & \textit{else} \end{cases} \quad (5)$$

Boundary Word Features: Following [14], we adopt boundary words as our basic features.

Document Topic Features: Given the document topic distribution inferred by LDA[1], we first choose the topic with the maximum probability to be the document topic. Then we use this topic as the document topic feature.

Word Topic Features: Topics of boundary words on the source side can be also used as features to capture topic-sensitive reordering patterns. Unfortunately, most of boundary words are function words or stop words (the percentage is 44.67% when we set the topic number 30). These words normally distribute evenly over all inferred topics and therefore their topics can not be used as indicators to distinguish topic-sensitive orientations. We believe that the integration of topics of boundary function words may jeopardize translation quality.

In order to address this issue, we use the topic assignments of content words that locate at the left/rightmost positions on the source phrases in question.

2.3 Training

In order to train the topic-based reordering model, topic distributions of source language documents from bilingual corpora are estimated by LDA tools. Train-

ing instances are generated and associated with their corresponding topic information. Given the features extracted from these training instances, we use a maximum entropy toolkit¹ to train the maximum entropy classifier (i.e., the topic-based reordering model). We perform 100 iterations of L-BFGS algorithm implemented in the training toolkit with both Gaussian prior and event cutoff set to 1 to avoid over-fitting.

3 Decoding

In this section, we integrate the proposed reordering model into SMT system. The log-linear model as described in [9] is adopted to combine various sub-models for obtaining the best translation, which is formulated as follows.

$$e_{best} = \operatorname{argmax}_e \left\{ \sum_{m=1}^M \lambda_m h_m(e, f) \right\} \quad (6)$$

where h_m are sub-models or features of the whole log-linear model, λ_m are their weights. We apply the topic-based reordering model as one sub-model to the log-linear model to constrain the phrase reordering, and tune the weight of the sub-model on development set.

We infer the topic distributions of the given test documents before translation according to the topic model trained on the corpora with document boundaries.

4 Experiments

In this section, we present our experiments on Chinese-to-English translation tasks with large-scale training data. With the aim of evaluating the effectiveness of the novel topic-based reordering model, we carried out a series of experiments with different configurations.

4.1 Setup

We adopted a state-of-the-art BTG-based phrasal system with a CKY-style decoder [14] as our baseline system and integrated the topic-based reordering model into this system.

Our training data consists of 2.8M sentence pairs with 81M Chinese words and 86M English words (including punctuation tokens) from LDC data². There are 10, 326 documents in the training data. We chose NIST MT Evaluation test set 2003 (MT03) as our development set, MT02, MT04 as our test sets. The number of documents/sentences in the development and test sets are listed in Table 1.

¹ Available at http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

² The corpora include LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08(Hong Kong Hansards/Law/News).

Table 1. The number of documents/sentences in the development set and test sets

	MT02	MT 03	MT04
The numbers of documents	100	100	200
The numbers of sentences	878	919	1788

Table 2. BLEU scores of the topic-based reordering model using document-level topic features on the development set with topic number K varying from 10 to 50

	DevSet(MT03)
Baseline	35.28
K = 10	35.48
K = 30	35.76
K = 50	35.39

We obtained the word alignments by running GIZA++ [9] on the training data in both directions and applying the “grow-diag-final-and” refinement [7]. We applied SRI Language Modeling Toolkit³ [8] to train a 5-gram model with Kneser-Ney smoothing on the Xinhua portion of English Gigaword corpus. Case-insensitive BLEU-4 [10] was used as our evaluation metric. In order to alleviate the impact of the instability of MERT, we run it three times for all our experiments and present the average BLEU scores on the three runs following the suggestion by [4].

To train the topic-based reordering model, we used the open source software tool GibbsLDA++⁴ for topic estimation and inference. GibbsLDA++ is an implementation of LDA using Gibbs Sampling technique for parameter estimation and inference. We removed Chinese stop words (1, 205 words in total) and rare words (54, 073 words in total) before topic estimation. Then we set the number of topic K, and used the default setting of the tool for training and inference. Document and word topic assignments for training sentences were obtained according to topic distributions. Finally, 52M training instances were generated to train the topic-based reordering model.

4.2 Impact of the Number of Topics

We carried out the first group of experiments to study the influence of the number of topics K on our topic-based reordering model. In these experiments, the topic-based reordering model was trained on the corpora with document boundaries. Only boundary words and document topic features are used in the trained reordering model. The results are shown in Table 2.

From Table 2, we can observe that we gain an improvement of 0.28 BLEU points when the number of topics K increases from 10 to 30. However a noticeable drop in the BLEU scores (-0.37 BLEU points) is observed, when K further

³ Available at <http://www.speech.sri.com/projects/srilm/download.html>

⁴ Available at <http://sourceforge.net/projects/gibbslda/>

Table 3. BLEU scores of topic-based reordering model with different features on the test sets (DT : document topic features; WT : word topic features)

	MT02	MT04	Avg
Baseline	33.90	34.86	34.38
Baseline + DT	34.11	35.52	34.82
Baseline + WT	34.09	35.05	34.57
Baseline + DT + WT	34.40	35.96	35.18

increases to 50. Therefore, the number of topics K is set to 30 in all experiments hereafter.

4.3 Effect of Different Topic Features

We ran our second group of experiments to investigate the effects of using different features in the topic-based reordering model. All reordering models were trained over the corpora with document boundaries. The reordering model of *Baseline* only uses boundary words as features.

The results of the topic-based reordering model compared with the conventional reordering model are presented in Table 3. As can be seen in the table, improvements are achieved by integrating topic features into the reordering model. Adding document topic features (+DT) and word topic features (+WT) leads to an average improvement of 0.44 BLEU points and 0.19 BLEU points respectively. Using all topic features(+DT+WT), our model achieves an improvement of up to 1.1 BLEU points on the MT04 test set, and an average improvement of 0.8 BLEU points over the *Baseline* on the two test sets. These improvements suggest that topic features are indeed useful for phrase reordering.

5 Analysis

In this section, we conduct a brief analysis of the results shown in the previous section, to take a deeper look into why the topic-based reordering model improves translation performance.

Table 4 shows an example which compares the translation generated by the *Baseline* to that generated by the *Baseline+DT+WT* system. By checking the translation process of the source sentence, we find that the baseline system incorrectly merges two phrases (“CEPA” and “to the legal profession in Hong Kong”) and makes a wrong reordering prediction. Under the guidance of topic information, our system avoids making the same error.

Furthermore, we view the reordering problem from a global perspective. The Chinese word “比/bǐ” which means “make a comparison” occurs frequently. The phrases to the right of the word “比/bǐ” are always moved to the left of the corresponding target word after translation. In other words, inverted orientation is often used when the source word “比/bǐ” occurs in documents with topic on economic performance comparison. However, when the word occur in documents

Table 4. A Chinese to English translation example showing the difference between the baseline and the system using our topic-based reordering model

src	(港/gǎng 澳/ào 台/tái) CEPA 为/wēi 香港/xiānggǎng 法律/fǎlǜ 界/jiè 打开/dǎkāi 新/xīn 局面/júmiàn
Baseline	(Hong Kong , Macao and Taiwan) to the legal profession in Hong Kong CEPA opens new situation
Baseline + DT + WT	(Hong Kong , Macao and Taiwan) CEPA opens new prospects to the legal profession in Hong Kong
ref	(Hong Kong , Macao and Taiwan) the CEPA opens up new prospects for Hong Kong legal fields

Table 5. Examples of different reordering phenomena in different topics, where phrases in brackets with the same index are translation of each other

Topic about economy	src	... (比/bǐ) ¹ (5月份/5yuèfèn) ² (下降/xiàjiàng 3.8%) ³ ...
	trg	... (down 3.8%) ³ (from) ¹ (May) ² ...
Topic about sport	src	... (五/wǔ) ⁴ (比/bǐ) ⁵ (-/yī) ⁶ ...
	trg	... (five) ⁴ (to) ⁵ (one) ⁶ ...

with a topic on sport, the inversion phenomena disappear because the word sense changes into “game score”.

Table 5 displays two examples where the word “比/bǐ” with different topics has different reorderings in bilingual data. This suggests that phrase reorderings are sensitive to topic and training instances with document-level topic features are capable of capturing topic-sensitive reordering patterns.

These examples and analysis empirically explain why reordering is topic-sensitive and our experimental results in Section 5 convincingly demonstrate that our proposed solution is effective in capturing topic information for phrase reordering.

6 Conclusions

In this paper, we have presented a topic-based reordering model to incorporate topic information into reordering model. To capture topic-sensitive reordering patterns, two kinds of topic information, namely the document-level topic and word-level topic, are used in our reordering model. Experimental results show that our model achieves substantial performance improvement over the baseline. This demonstrates the advantage of exploiting topic information for phrase reordering. Finally, we investigate why incorporating topic information into reordering model improves SMT performance.

In future, instead of only using the document topic with the maximum probability, we would like to utilize topic distribution to represent document topic features. Furthermore, we plan to explore more document-level information to predict orders for neighboring blocks. Finally, we are also interested in investigating new methods to infer topics for sentences with no document boundaries.

Acknowledgement. The authors are supported by National Natural Science Foundation of China (No. 61373095,61373097,61272259) and Natural Science Foundation of Jiangsu Province (No. BK20140355).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *The Journal of Machine Learning Research*, 601–608 (2002)
2. Chen, B., Foster, G., Kuhn, R.: Adaptation of reordering models for statistical machine translation. In: *Proceedings of NAACL-HLT*, pp. 938–946 (2013)
3. Clark, J.H., Dyer, C., Lavie, A., Smith, N.A.: Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In: *Proceedings of ACL*, pp. 176–181 (2011)
4. Gong, Z., Zhang, M., Zhou, G.: Cache-based document-level statistical machine translation. In: *Proceedings of EMNLP*, pp. 909–919 (2011)
5. Gong, Z., Zhang, M., Tan, C., Zhou, G.: N-gram-based tense models for statistical machine translation. In: *Proceedings of EMNLP*, pp. 276–285 (2012)
6. Knight, K.: Decoding complexity in word-replacement translation models. *Computational Linguistics* 25(4), 607–615 (1999)
7. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: *Proceedings of NAACL-HLT*, pp. 48–54 (2003)
8. Stolcke, A.: SRILM-an extensible language modeling toolkit. In: *Proceedings of INTERSPEECH* (2002)
9. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: *Proceedings of ACL*, pp. 311–318 (2002)
11. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics* 23(3), 377–403 (1997)
12. Xiao, X., Xiong, D., Zhang, M., Liu, Q., Lin, S.: A topic similarity model for hierarchical phrase-based translation. In: *Proceedings of ACL*, pp. 750–758 (2012)
13. Xiong, D., Ben, G., Zhang, M., Lü, Y., Liu, Q.: Modeling lexical cohesion for document-level machine translation. In: *Proceedings of IJCAI*, pp. 2183–2189 (2013)
14. Xiong, D., Liu, Q., Lin, S.: Maximum entropy based phrase reordering model for statistical machine translation. In: *Proceedings of ACL*, pp. 521–528 (2006)
15. Xiong, D., Zhang, M.: A topic-based coherence model for statistical machine translation. In: *Proceedings of AACL*, pp. 2183–2189 (2013)

Online Chinese-Vietnamese Bilingual Topic Detection Based on RCRP Algorithm with Event Elements

Wen-xu Long^{1,2}, Ji-xun Gao³, Zheng-tao Yu^{1,2,*},
Sheng-xiang Gao^{1,2}, and Xu-dong Hong^{1,2}

¹ School of Information Engineering and Automation,
Kunming University of Science and Technology, Kunming 650051, China

² The Intelligent Information Processing Key Laboratory,
Kunming University of Science and Technology, Kunming 650051, China

³ School of Computer Science, Henan Institute of Engineering, Zhengzhou, 451191, China

Abstract. On account of the characteristics of online Chinese-Vietnamese topic detection, we propose a Chinese-Vietnamese bilingual topic model based on the Recurrent Chinese Restaurant Process and integrated with event elements. First, the event elements, including the characters, the place and the time, will be extracted from the new dynamic bilingual news texts. Then the word pairs are tagged and aligned from the bilingual news and comments. Both the event elements and the aligned words are integrated into RCRP algorithm to construct the proposed bilingual topic detection model. Finally, we use the model to determine if the new documents will be grouped into a new category or classified into the existing categories, as a result, to detect a topic. Through the contrast experiment, the proposed model achieves a good effect on topic detection.

Keywords: Topic model, Event elements, Storyline. Bilingual, RCRP.

1 Introduction

Vietnam is closely connected with China so that to timely and accurately detect Chinese-Vietnamese bilingual topic trend is of great significance to enhance the communication and cooperation between both sides. On monolingual topic detection, a large number of research has been done at home and abroad. Considering of the elements of news, Wang extracted the name entities and integrated them into LDA model to track a series of related news[1]. On bilingual topic detection, De Smet W. proposed an intermediate LDA model of English and Dutch, which were trained from English-Dutch word pairs of Wikipedia[2]. Ni, et al, proposed a cross language classification model by minging multilingual topics from Wikipedia page and data[3]. Considering of the dynamic topic detection, Ahmed integrated RCRP algorithm with LDA model according to the temporal of the news, which gained good effect[4-6].

Online Chinese-Vietnamese topic detection is to analysis the dynamic growing bilingual news text. Hence According to the characteristics of news, we need to combine the key elements of an event of who, when and where and analysis the text relevance by the constructed entities, eg. who, when and where; Also it should timely acquire the

growing mixed bilingual news data and be able to analysis the data dynamical-ly;Bilingual news has the characteristic of cross-language and we need reduce the error from direct translation. The core of the RCRP algorithm is a nonparametric Bayesian method,using the prior parameters of this time to estimate parameters of next period,from which can provide a dynamic analysis in constant periods.RCRP has been a periodic and nonparametric evolving clustering method, which can file a new document into the existing cluster according to a prior probability. Meanwhile it is featured with the unfixed number of clusters and can get a cluster at any time based on a specific probability,which accord with the characteristic of the topic of randomly generated and developing,extincting with time[5-6]. Hence that according to the characteristics of online bilingual topic detection and the advantage of RCRP,we try to exploit a model based on RCRP algorithm,which integrates the temporal information, the entities and bilingual aligned words to solve the topic detection problem.

2 Online Bilingual LDA Integrated with Event Elements

2.1 To Integrate Time Series with RCRP

As a special case in the Dirichlet process, the recurrent Chinese restaurant process is on the basis of the LDA model[5-6] and according to the Markov Assumption, assume the parameters $\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \delta^2 I)$ and $\beta_t|\beta_{t-1} \sim N(\beta_{t-1}, \delta^2 I)$, instead of assuming that α and β would remain unchanged at any point of time within a certain time frame.

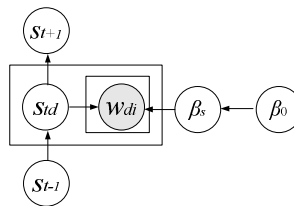


Fig. 1. RCRP algorithm graphical model

For time $t \in \{1, 2, \dots, T\}$

- (a) Get s_{td} by $s_{td}|s_{1:t-1}, s_{t,1:d-1}$
- (b) If s_{td} is a new storyline get $\beta_s|\beta_0$
- (c) Get $w_{di} \sim \beta_0$ for news text d_i

2.2 To Integrate Event Elements

After an event has happened, various reports about this event will be made from every aspect. Although the words used and the opinions expressed in each report are different, a consensus is always reached on the key questions connected to the event. In fact, all of the event elements such as the time and entities including the name of the person, the name of the place and the name of the organization have become an im-

portant approach for the differentiation of news topics with different plots, which will be extracted from the news text.

In LDA model[7], the topic distribution in a document is subject to the wording condition and obtained through the calculation of the word frequency. But some words are useless or even bring deviation for topic detection. Therefore it's necessary to use the entities as the label information for the detection work.

2.3 To Integrate Aligned Bilingual Event Arguments

Huge work has been done on the fields of bilingual entity translation[8], word alignment[9] and cross-language entity linking[10]. With the methods of these work and through the Chinese-Vietnamese page from Wikipedia, it's possible to get the comparable corpus. For word-to-word translation from Chinese V_s to Vietnamese V_t with m_D defined as (v_s, v_t) and $v_s \in V_s, v_t \in V_t$. And it's applicable to utilize the ontology-based event knowledge base to calculate the semantic similarity between the nouns, the verbs and the entities in the bilingual documents, selecting those pairs with high similarity as the collection of synonym pairs, m_K . Meanwhile with the application of the alignment method for the bilingual event elements contained in the news, the aligned elements will be obtained from the bilingual pages to constitute the collection of aligned bilingual event arguments, m_E . The total collection $m = m_D \cup m_K \cup m_E$, which is inputted into LDA to construct a bilingual topic model. We train Chinese and Vietnamese data separately to get each posterior parameter γ and π , and get a joint distribution by the collection m .

2.4 The Proposed Online Bilingual LDA Integrated with Event Elements

As shown in Fig.2., it is probabilistic graphical model constructed for the Chinese-Vietnamese topics.

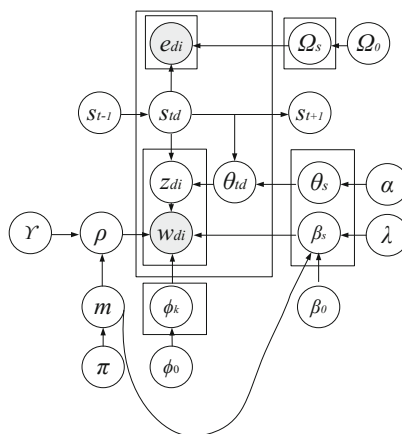


Fig. 2. Online bilingual LDA integrated with event elements

We use Gibbs Sampling to train the model using the joint distribution of bilingual information. The joint distribution is gotten from the algorithm below.

- (a) Setting the topic number is K ;
- (b) Setting the parameter α_0 and ϕ_0 ;
 Sampling K times $\gamma \sim Dirichlet(\phi)$
 Sampling K times $\pi \sim Dirichlet(\phi)$
- (3) For each $d \in \{d_1, \dots, d_t\}$
 Sampling $\rho \sim Dirichlet(\alpha)$
- (4) Draw the story indicator:
 $s_{td} | s_{1:t-1}, s_{t,1:d-1} \sim RCRP(\gamma, \lambda, \Delta)$
- (5) If s_{td} is a new story,
 Sampling K times over aligned words $\beta_{s_{new}} \sim Dirichlet(\beta_0)$
 Sampling K times over entities $\Omega_{s_{new}} \sim Dirichlet(\Omega_0)$
 Sampling K times over entities $\theta_{s_{new}} \sim Dirichlet(\alpha)$
 Topic proportions $\theta_{td} \sim Dirichlet(\theta_{std})$
 Entities $e_{td} \sim Multinomial(\Omega_{std})$

Through the calculation of the posterior probability, $P(z_{1:T}, s_{1:T} | x_{1:T})$, we shall manage to realize the Chinese-Vietnamese topic detection with z_t, s_t, x_t representing separately the subject marker, the topic marker and the bag of words, which consists of the event elements w_{td}^{K+1} that cover the aligned bilingual entity e_{td} and the time etc. in the time slot of t . In a new document corresponding to the event s , the probability based on which x_t will be assigned with the i th topic among the existing $K-1$ topics within the time of t is shown as below:

$$P(z_{td} = z_{tdi} | s_{ts}^{-td}, x_{td}, rest) = \frac{C_{tdi}^{-i} + \frac{C_{si}^{-i} + \alpha}{C_s^{-i} + \alpha(K+1)}}{C_{td.}^{-i} + 1} \frac{C_{ix} + \phi_0}{C_i + \phi_0 W} \tag{1}$$

Herein, C_{tdi}^{-i} refers to the number of the topics without the i th topic in the document d at the time, t . C_{sk}^{-i} represents the number of the topics corresponding to the event s without the i th topic, while C_{kxt}^{-i} stands for the number of the topics covered with the word x without the i th topic.

$$C_{td.}^{-i} = C_{tdi}^{-i}, s. - i = C_{si}^{-i}, C_i = C_{ix} \tag{2}$$

For the model we propose,

$$P(s_{td} | s_{t-\Delta:t}^{-td}, x_{td}, rest) = P(s_{td} | s_{t-\Delta:t}^{-td}) P(s_{td} | s_{t-\Delta:t}^{-td}, rest) P(z_{td} | s_{td}, rest) P(e_{td} | s_{td}, rest) P(w_{td}^{K+1} | s_{td}, rest) \tag{3}$$

Where $rest$ denotes all other hidden variables not including z_t, s_t, x_t .

The posterior probability, P is computed using the chain rule as follows:

$$P(z_{1:T}, s_{1:T} | x_{1:T}) = \prod_{i=1}^{n_{td}} P(z_{tdi} | s_{td} = s, z_{td}^{-td}, rest) \quad (4)$$

3 Experiments and Analyses

3.1 Evaluation Index

Regarding the clustering performance, generally tests have been conducted and the detection error cost, C_{det} has been applied to evaluate the effect and performance of the algorithm. Consisting of the miss rate P_{miss} of the model and the false detecting rate, P_{fa} , C_{det} has been considered as the evaluation criteria published by the National Institute of Standards and Technology (NIST) for the topics and tasks with the specific calculations described as below.

$$C_{det} = C_{miss} \times P_{miss} \times P_{target} + C_{fa} \times P_{fa} \times P_{non-target} \quad (5)$$

Herein, C_{miss} represents the cost coefficient of the missing detection with C_{fa} standing for the cost coefficient of the false detection, while P_{target} means the prior probability for the system to make positive judgment with $P_{non-target}$ representing the prior probability for the system to make negative judgment. However according to the standard made by NIST, we generally assume that C_{miss} and C_{fa} are separately 1 and 0.1 with the values for P_{target} and $P_{non-target}$ respectively being 0.02 and 0.98. Below please find the calculation formula of P_{miss} and P_{fa} with the parameters defined in the table as below.

During the application, generally evaluation is made according to the normalized detection error cost with the calculation formula given as below:

$$Norm(C_{det}) = \frac{C_{det}}{\min(C_{miss} \times P_{target}, C_{fa} \times P_{non-target})} \quad (6)$$

3.2 Results and Analyses

In the experiment, 376274 news and reports have been fetched from the Chinese website and 221035 reports have been grabbed from the Vietnamese website to act as the data set. Half of the data and Wikipedia data is used to train the model. According to the web page tags, we could find that the topic, the release time and the content of the reports have been covered completely in the data acquired for the news. In fact, the experiment has been conducted for the following purposes on the basis of the dataset: (1) Observe the detection error cost for the model when the number of topics is different; (2) Observe the detection error cost when separately removing the event element, the time sequence and the bilingual word pairs from the model; (3) In the case that the number of the topics have been specified, make a comparison with the K-means clustering algorithm and the LDA model that hasn't been integrated with the features. For

the hyper-parameters of the model, make sure that $\beta_0 = 0.1$, $\phi_0=0.01$, $\Omega_0=0.001$, $\alpha_0 = \frac{0.1}{K+1}$, $\lambda = 0.5$ and $\Delta = 3$.

- Comparison of different assumpt topic numbers

In the experiment, we’ve also tested the effect of the model in the case of different number of topics K by assuming that the number of Gibbs sampling is N=500.

Table 1. The evaluation of the model in the case of different topic number

K	1	30	50	100	200	300
C_{det}	0.79	0.78	0.740	0.697	0.733	0.738

Judging from the table as above, we could find that when K=100, the value of C_{det} is minimum and slow changes of the value of C_{det} have been found when $K<50$ or $K>100$. All of these prove that in a given dataset, there should be an appropriate K that might lead to the minimum C_{det} in the model. In fact, C_{det} of the model will be reduced with the increase in the number of the topics within a certain range. However when the number of the topics reaches a threshold, the influence on the model brought by the increase in the number will become weaker and weaker.

- With different features(event elements、storyline and bilingual word pairs)

In order to illustrate the role and importance of the three model features which have been integrated : the event elements, the time series and the collection of bilingual word pairs, we’d like to remove a certain element from the model for the contrast experiment in the test for the purpose to test the influence of each element on the model. Meanwhile in the test, we’ve determined that the number of the topic, K=100 and the sampling number N=500.

Table 2. The evaluation of the model in the case of integrating different features

Features	C_{det}
Storyline and bilingual word pairs	0.91
Event elements and bilingual word pairs	0.733
Event elements and Storyline	0.79

According to the result, we could find that the event elements have played a critical role in the topic detection. However the collection of bilingual word pairs has contributed a lot to the improvement of the model effect due to the reduction in the error caused by the polysemy when various translation tools are utilized.

- Compared with K-means clustering and simple LDA

In the test, first on account of the 1000 Chinese and Vietnamese news about the anti-Chinese movements in Vietnam and the other kinds of bilingual news with the amount of news up to 1000, we shall evaluate the performance of these three models regarding the anti-Chinese movements in Vietnam. During the application of K-means algorithm, we assume that the clustering number, K is set as 20, 30 and 50. While for the LDA model, we've assumed that the number of the topics is 100 with the sampling number at 300.

Table 3. The evaluation of different algorithm on topic detection

Algorithm		C_{det}
K-means	K=20	0.861
	K=30	0.847
	K=50	0.851
Simple LDA		0.89
Proposed model		0.714

Through the aforesaid tests, we could find that the online bilingual LDA model integrated with the event elements has been superior to the other two kinds of models/algorithms from the perspective of the evaluation result. Limited by the computation process, the K-means algorithm is not applicable to massive dataset, the unknown clustering center and the changes in the incremental data. However if the LDA hasn't been integrated with the features, it means that the event elements haven't been covered in the model to bring noises to numerous words in the bag of words.

3.3 Experimental Evaluation

According to the test result, it turns out that: the event elements and the time sequence contained in the news and reports and the creation of bilingual words have influenced a lot the effect of the topic detection. The proposed model, which has been developed on the basis of the traditional LDA model and integrated with the event elements, the time series and the Chinese-Vietnamese word pairs, is able to cluster more accurately the data of the bilingual news.

4 Conclusion

In this paper, we propose a RCRP-based online Chinese-Vietnamese topic detection model according to the characteristics of dynamic bilingual news, which effectively integrates time series, event elements and bilingual information into one LDA model and achieves good effect in the experiment. Our work next step is to exploit a more advanced topic model by using bilingual machine translation and bilingual knowledge

resources to calculate the relativity of news on text level to implement the news clustering.

Acknowledgments. This paper is supported by National Nature Science Foundation (No.61472168,61262041,61175068), and the key project of Yunnan Nature Science Foundation (2013FA030),and Science and technology innovation talents fund projects of Ministry of Science and Technology(No.2014HE001) .

References

1. Wang, D., Liu, W., Xu, W.: Topic Tracking Based on Event Network. In: 2011 4th International Conference on Cyber, Physical and Social Computing Internet of Things (iThings/CPSCoM), pp. 488–493 (2011)
2. De Smet, W., Moens, M.F.: Cross-language linking of news stories on the web using interlingual topic modelling. In: Proceedings of the 2nd ACM Workshop on Social Web Search and Mining, pp. 57–64. ACM (2009)
3. Ni, X., Sun, J.-T., Hu, J., Chen, Z.: Cross Lingual Text Classification by Mining Multilingual Topics From Wikipedia. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 375–384. ACM (2011)
4. Ahmed, A., Xing, E.P.: Dynamic Non-parametric Mixture Models and the Recurrent Chinese Restaurant Process: With Applications to Evolutionary Clustering. In: SDM (2008)
5. Ahmed, A., Ho, Q., Eisenstein, J., et al.: Unified analysis of streaming news. In: Proceedings of the 20th International Conference on World Wide Web, pp. 267–276. ACM (2011)
6. Ahmed, Q., Ho, C., Teo, J., Eisenstein, A.J., Smola, E.P.: Xing The Online Infinite Topic-Cluster Model: Storylines From Streaming Text. CMU-ML-11-100 (2011)
7. Blei, D.M., Andrew, Y.N., Michael, I.J.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
8. Sproat, R., Tao, T., Zhai, C.X.: Named Entity Transliteration with Comparable Corpora. In: Proceeding ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 73–80 (2006)
9. Espla-Gomis, M., Sanchez-Martinez, F., Forcada, M.L.: A Simple Approach to Use Bilingual Information Sources for Word Alignment. *Procesamiento del Lenguaje Natural*, 93–100 (2012)
10. Fahrni, A., Strube, M.: HITS' Cross-lingual Entity Linking System at TAC 2011:One Model for All Languages. In: Proceeding of Text Analysis Conference, November 14-15 (2011)

Random Walks for Opinion Summarization on Conversations

Zhongqing Wang, Liyuan Lin, Shoushan Li, and Guodong Zhou

Natural Language Processing Lab, School of Computer Science and Technology,
Soochow University, China

{wangzq.antony, scarecrowlly, shoushan.li}@gmail.com,
gdzhou@suda.edu.cn

Abstract. Opinion summarization on conversations aims to generate a sentimental summary for a dialogue and is shown to be much more challenging than traditional topic-based summarization and general opinion summarization, due to its specific characteristics. In this study, we propose a graph-based framework to opinion summarization on conversations. In particular, a random walk model is proposed to globally rank the utterances in a conversation. The main advantage of our approach is its ability of integrating various kinds of important information, such as utterance length, opinion, and dialogue structure, into a graph to better represent the utterances in a conversation and the relationship among them. Besides, a global ranking algorithm is proposed to optimize the graph. Empirical evaluation on the Switchboard corpus demonstrates the effectiveness of our approach.

Keywords: Opinion Summarization on Conversations, Graph, Random Walk, Global Ranking.

1 Introduction

Opinion summarization aims to generate a sentimental summary on opinions in a text and has been drawing more and more attention recently in NLP due to its significant contribution to various real applications [5, 14]. However, although there are a few previous studies on extracting opinion summaries, most of them focus on text reviews, such as movie reviews [8, 16] and product reviews [4, 21]. With the increasing amount of conversation recordings, opinion summarization on conversations becomes more and more demanding. In this study, we investigate this novel type of opinion summarization.

Speech summarization is more difficult than well-structured text, because a) speech is always less organized and has recognition errors; b) in conversational speech, information density is low and there are often off topic discussions [24].

As pilots in opinion summarization on conversations, Wang and Liu [24] recast it as an utterance ranking problem, similar to traditional topic-based summarization and general opinion summarization [18, 23]. However, as stressed by Wang and Liu [24], opinion summarization on conversations possesses some unique characteristics and challenges.

First, it is necessary to consider both the topic relevance of an utterance and the opinion expressions in an utterance. One basic intuition therein is that opinion summarization is prone to containing opinion sentences.

Second, dialogue structures play an important role in utterance selection. One common phenomenon is that if one utterance contains opinions, its adjacent paired utterances much likely contain opinions.

Third, there may be some short utterances in a conversation, e.g. “Uh”, “Yeah”, “Well”, etc. Our preliminary exploration finds that, although most of them are little informative, they are much likely to be selected as “good” utterance candidates in the summary when some frequency-based approaches are employed.

Although above unique characteristics and challenges have been noticed by Wang and Liu [24], they are not well addressed in the literature. This largely limits the performance of opinion summarization on conversations.

In this paper, we re-visit these unique characteristics and challenges, and propose a graph-based framework to opinion summarization on conversations. In particular, a random walk model is proposed to globally rank the utterances in a conversation. The main advantage of our approach is its ability of integrating various kinds of important information, such as utterance length, opinion, and dialogue structure, into a graph to better represent the utterances in a conversation and the relationship among them. Different from Wang and Liu [24], where these factors are separately ranked and combined with a simple weighting strategy, we incorporate them into an utterance graph and perform global graph ranking. Experimental results on the Switchboard corpus show that our approach achieves the performance of 0.5778 in terms of ROUGE-1 measurement, which is 0.034 higher than that reported in Wang and Liu [24].

The rest of this paper is organized as follows. Section 2 overviews the related work on both topic-based summarization and opinion summarization. Section 3 introduces our framework for opinion summarization on conversations. Section 4 reports the experiment results. Finally, Section 5 concludes this paper with future work.

2 Related Work

Some previous studies on opinion summarization focus on to generate aspect-based ratings for an entity [4, 21] which actually consider the opinion summarization as an opinion mining problem. Although such summaries are informative, they lack critical information for a user to understand why an aspect receives a particular rating. Ganesan et al. [5] present a graph-based summarization framework named *Opinosis* to generate concise abstractive summaries of highly redundant opinions. Nishikawa et al., [14] generates a summary by selecting and ordering sentences taken from multiple review texts according to represent the informative and readability of the sentence order.

To the best of our knowledge, there is only one related work on opinion summarization on conversation, i.e., Wang and Liu [24]. They create a corpus containing both extractive and abstractive summaries of speaker’s opinion towards a given topic using telephone conversations. They adopt both sentence-ranking method and graph-based method to perform extractive summarization. However, since they consider the topic,

opinion, dialogue structure and sentence length factors separately, the relations of them are not well integrated. Unlike that, our approach leverages them together in a random walk model and makes them working together to improve the overall performance.

3 Random Walks for Opinion Summarization on Conversations

Formally, a conversation is denoted as its contained utterances as $U = [u_1, u_2, \dots, u_n]^T$ and the summary using the extracted utterances as $X = [x_1, x_2, \dots, x_m]^T$ with $m < n$ and $X \subset U$. Our approach for opinion summarization on conversations consists of three main steps: First, we build a graph G to represent all the utterances with their mutual topic, opinion and dialogue structure information; second, we rank the utterances on graph G with PageRank algorithm. Finally, we select the utterances with top ranking scores to generate a summary.

3.1 Graph Building

Different from traditional topic-based summarization tasks [18, 22], opinion summarization on conversations is encouraged to consider not only the topic relevance, but also the opinion and dialogues structure factors [24]. To integrate them into a uniform graph-based ranking framework, we hope to build a graph which could include all these information.

To achieve that, we build a graph that contains topic relevance, opinion relevance and dialogue structure relevance, which makes the graph a tri-layer model. The first layer contains the topic information; the second one contains the opinion information; the third one contains dialogue structure information.

Representing an Utterance as a Feature Vector

In our approach, an utterance is considered as a node in the graph and it is represented by a feature vector. If two sentences are more related, their feature vectors are supposed to share more features. To represent the relationship between two utterances, three kinds of features are employed to represent topic relevance, opinion relevance and structure relevance respectively.

- **Topic Relevance Features:** If two utterances shares more word unigrams and bigrams, they are thought to be more topic-related, as popularly assumed by many other previous studies [22]. Thus, the word unigrams and bigrams are adopted as the features to representing the topic relevance. The weight of each feature is Boolean which represents the presence or absence of a feature in an utterance. Formally, an utterance u_i can be represented as a feature vector x_i ,

$$u_i = \langle \text{bool}(t_1), \text{bool}(t_2), \dots, \text{bool}(t_n) \rangle$$

Where n is the number of unique features; $\text{bool}(t_i) = 1$ means the occurrence of the feature t_i in the utterance and $\text{bool}(t_i) = 0$ means the absence of the feature t_i .

- **Opinion Relevance Features:** If two utterances both contain opinion, they are thought to be opinion-related. To represent such relationship, a new feature named OPINION is added for each utterance. If an utterance contains at least one sentiment word in the pre-given lexicon, the OPINION feature weight is set to be a fixed integer larger than one, i.e., $\lambda (\lambda > 1)$; Otherwise, the feature weight is set to zero. In this study, the sentimental lexicon¹ we used is from MPQA.
- **Structure Relevance Features:** There are two dialogue structures in conversations: One is the *adjacent* relation representing the relation between two adjacent utterances which are said by two speakers; the other is the *turn* relation representing the relation between two utterances which are in the same turn. If two utterances take either the *adjacent* relation or the *turn* relation, they are considered to be more structure-related. To represent such relationships, two kinds of features named ADJ and TURN are added for each utterance. Specifically, 1) we let two adjacent utterances u_i and u_{i+1} share the same feature ADJ- i ; 2) we let two utterances u_i and u_j in the same turn k share the same feature TURN- k . Because these two kinds of features are believed to be more important than one unigram word feature, their weights are set to be a fixed integer that larger than one, i.e., $\omega (\omega > 1)$.

Transition Probability Computation with the Penalization on Short Utterances

The transition probability from the i -th node to the j -th node, denoted as $p(i \rightarrow j)$ is defined as the normalization of the weights of the edges out of the i -th node, i.e.,

$$p(i \rightarrow j) = \frac{f(i \rightarrow j)}{\sum_k f(i \rightarrow k)}$$

Where $f(i \rightarrow j)$ represents the similarity between u_i and u_j . Here, the cosine similarity is adopted (Baeza-Yates and Ribeiro-Neto, 1999):

$$f(i \rightarrow j) = \frac{u_i \cdot u_j}{|u_i| |u_j|}$$

Two nodes are connected if their transition probability is larger than zero, i.e., $p(i \rightarrow j) > 0$. To avoid self-transition, we set the transition probability of one node to itself as zero.

Short utterances, e.g., backchannels, appear frequently in dialogues while they typically contain little important content [24]. For example, as shown in Fig 3, we can see that the short utterances such as u_2 (*and*) and u_5 (*Oh*) contains little information for opinion expression.

Since the short utterances are sometimes highly frequency words, and thus the transfer probability between these utterances are larger than many other utterances, which makes the PageRank algorithm more likely to select long utterances. To avoid this happening, we propose a novel formula of similarity function $f(i \rightarrow j)$ to penalize the

¹ http://www.cs.pitt.edu/mpqa/subj_lexicon.html

short utterances using the information of the utterance length. The basic idea is to make the transition probability to be proportion to the length of the utterance. In this way, the shorter the utterance, the lower transition probability to it will be assigned. The revised formula for computing transition probability is given as follows:

$$f(i \rightarrow j) = \frac{u_i \cdot u_j}{|u_i| |u_j|} \log(|u_j|)$$

3.2 Ranking the Utterances with PageRank

Given the graph G , the saliency score $s(u_i)$ for utterance u_i can be deduced from those of all other utterances linked with it and it can be formulated in a recursive form as in the standard PageRank algorithm.

$$s(u_i) = \mu \sum_{j \neq i} s(u_j) \cdot p(j \rightarrow i) + (1 - \mu)$$

In the implementation, μ is the damping factor and usually set to be 0.85 (Page et al., 1998). The initial scores of all utterances are set to one, and the iteration algorithm is adopted until convergence [22].

As long as the saliency scores of utterances are obtained, the utterances are ranked with the scores. The utterances with largest ranking scores form the summary. In the implementation, the utterances from both speakers in the conversion are emerged for ranking with our PageRank algorithm.

4 Experimental Evaluation

4.1 Evaluation Setup

In the experiment, we use the Opinion Conversion Corpus which is drawn from the Switchboard corpus [24]. The corpus contains 88 conversations from 6 topics, among which 18 conversations are annotated by three annotators. We use these 18 annotated conversations as the testing set and perform our ranking approach on it.

We use the ROUGE toolkit [10] which has been widely adopted for automatic summarization evaluation. We choose three automatic evaluation methods ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-W in our experiment.

4.2 Experimental Results

In this subsection, we evaluate the performance of our PageRank approach and compare it with three baseline approaches for opinion summarization on conversions, together with the human summarization:

- **Max-length:** select the longest utterances for each speaker, this has been shown to be a very strong baseline for summarization on conversations (Gillick et al., 2009).
- **Sentence-Ranking:** the topic, opinion and dialogue structure score are separately calculated and then combined via a linear combination (Wang and Liu, 2011). We report the best performance of ROUGE-1 measurement for comparison.
- **PageRank:** the PageRank approach which only considers the topic relevance, which is a popular approach in topic-based text summarization (Wan and Yang, 2008).
- **Human:** calculate ROUGE scores between each reference and the other references, and average them. This can be considered as the upper bound of the performance.

Followed by Wang and Liu [24], the average compression ratio of the extractive summary of each conversation is set to be 0.25. In our approach, we set the parameters as $\lambda = 6$ and $\omega = 5$.

Table 1 shows the results of different approaches. From this table, we can see that the approach by Wang and Liu [24] is more effective than the basic PageRank approach, i.e., Topic-PageRank. This is because it takes the specific characteristics in conversations, such as utterance length and opinion information, into account. Our approach outperforms all the other approaches and improves the performances from 0.5448 to 0.5778 compared to the approach by Wang and Liu [24].

Table 1. Comparison Results with Baseline Approaches

Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
Max-length	0.5279	0.3600	0.5113	0.2495
Sentence-Ranking	0.5448	-	-	-
PageRank	0.4959	0.3105	0.4821	0.2293
Our Approach	0.5778	0.4202	0.5670	0.2786
Human	0.6167	0.5200	0.6100	0.3349

It is interesting to find that the simplest baseline Max-length is able to get a decent performance of 0.5279, which is even much better than Topic-PageRank. This result reveals that the length of the utterances is an important factor for selecting “good” utterances in summarization on conversations.

5 Conclusion and Future Work

In this paper, we propose a graph-based framework to opinion summarization on conversations by first representing a conversation as an utterance graph and then performing global ranking via a PageRank algorithm. Besides topic relevance, both opinion relevance and structure relevance are incorporated systematically to meet the

specific characteristics and challenges in the task. Empirical studies demonstrate that our approach performs much better than other alternatives.

The research of opinion summarization on conversations is still in its early stage since the pilot work by Wang and Liu [24]. In the future work, we will explore more factors in a conversation and better ways of representing a conversation.

Acknowledgments. This research work is supported by the National Natural Science Foundation of China (No. 61273320, No. 61331011, and No. 61375073), National High-tech Research and Development Program of China (No. 2012AA011102). We thank Dr. Dong Wang for providing their corpus and useful suggestions. We also thank anonymous reviewers for their valuable suggestions and comments.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. ACM Press and Addison Wesley (1999)
2. Celikyilmaz, A., Hakkani-Tur, D.: Discovery of Topically Coherent Sentences for Extractive Summarization. In: *Proceeding of ACL 2011* (2011)
3. Erkan, G., Radev, D.: LexPageRank: Prestige in Multi-document Text Summarization. In: *Proceedings of EMNLP 2004* (2004)
4. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: *Proceedings of SIGKDD 2004* (2004)
5. Ganesan, K., Zhai, C., Han, J.: Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In: *Proceeding of COLING 2008* (2008)
6. Koumpis, K., Renals, S.: Automatic Summarization of Voicemail Messages using Lexical and Prosodic Features. *ACM-Transactions on Speech and Language Processing* (2005)
7. Li, F., Tang, Y., Huang, M., Zhu, X.: Answering Opinion Questions with Random Walks on Graphs. *Proceeding of ACL 2010* (2010)
8. Li, S., Huang, C., Zhou, G., Lee, S.: Employing Personal/Impersonal Views in Supervised and Semi-supervised Sentiment Classification. In: *Proceedings of ACL 2010* (2010)
9. Lin, C.: Training a Selection Function for Extraction. In: *Proceedings of CIKM 1999* (1999)
10. Lin, C.: ROUGE: A Package for Automatic Evaluation of Summaries. In: *Proceedings of ACL 2004, Workshop on Text Summarization Branches Out* (2004)
11. Lin, S., Chen, B., Wang, H.: A Comparative Study of Probabilistic Ranking Models for Chinese Spoken Document Summarization. *ACM Transactions on Asian Language Information Processing* 8(1) (2009)
12. Mckeown, K., Hirschberg, J., Galley, M., Maskey, S.: From Text to Speech Summarization. In: *Proceedings of ICASSP 2005* (2005)
13. Murray, G., Carenini, G.: Detecting Subjectivity in Multiparty Speech. In: *Proceedings of Interspeech 2009* (2009)
14. Nishikawa, H., Hasegawa, T., Matsuoand, Y., Kikui, G.: Optimizing Informativeness and Readability for Sentiment Summarization. In: *Proceeding of ACL 2010* (2010)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Libraries (1998)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: *Proceedings of EMNLP 2002* (2002)

17. Radev, D., McKeown, K.: Generating Natural Language Summaries from Multiple Online Sources. *Computational Linguistics* 24(3), 469–500 (1998)
18. Radev, D., Jing, H., Stys, M., Tam, D.: Centroid-based Summarization of Multiple Documents. *Information Processing and Management* 40, 919–938 (2004)
19. Raaijmakers, S., Truong, K., Wilson, T.: Multimodal Subjectivity Analysis of Multiparty Conversation. In: *Proceedings of EMNLP 2008* (2008)
20. Ryang, S., Abekawa, T.: Framework of Automatic Text Summarization Using Reinforcement Learning. In: *Proceeding of EMNLP 2012* (2012)
21. Titov, I., Mc-donald, R.: A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In: *Proceedings of ACL 2008* (2008)
22. Wan, X., Yang, J.: Multi-document Summarization using Cluster-based Link Analysis. In: *Proceedings of SIGIR 2008* (2008)
23. Wan, X.: Using Bilingual Information for Cross-Language Document Summarization. In: *Proceedings of ACL 2011* (2011)
24. Wang, D., Liu, Y.: A Pilot Study of Opinion Summarization in Conversations. In: *Proceeding of ACL 2011* (2011)
25. Xie, S., Liu, Y.: Improving Supervised Learning for Meeting Summarization using Sampling and Regression. *Computer Speech and Language* 24, 495–514 (2010)
26. Zhang, J., Chan, H., Fung, P.: Improving Lecture Speech Summarization using Rhetorical Information. In: *Proceedings of Biannual IEEE Workshop on ASRU* (2007)
27. Zhu, X., Penn, G.: Summarization of Spontaneous Conversations. In: *Proceedings of Interspeech 2006* (2006)

TM-ToT: An Effective Model for Topic Mining from the Tibetan Messages

Chengxu Ye^{1,2,*}, Wushao Wen^{1,3}, and Ping Yang⁴

¹ School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China

² School of Computer Science, Qinghai Normal University, Xining 810008, China

³ School of Software, Sun Yat-sen University, Guangzhou 510006, China

⁴ School of Life and Geography Sciences, Qinghai Normal University, Qinghai Xining 810008, China

ycx@qhnu.edu.cn, wenwsh@mail.sysu.edu.cn, ypycx@163.com

Abstract. The microblog platforms, such as Weibo, now accumulate a large scale of data including the Tibetan messages. Discovering the latent topics from such huge volume of Tibetan data plays a significant role in tracing the dynamics of the Tibetan community, which contributes to uncover the public opinion of this community to the government. Although topic models can find out the latent structure from traditional document corpus, their performance on Tibetan messages is unsatisfactory because the short messages cause the severe data sparsity challenge. In this paper, we propose a novel model called TM-ToT, which is derived from ToT (Topic over Time) aiming at mining latent topics effectively from the Tibetan messages. Firstly, we assume each topic is a mixture distribution influenced by both word co-occurrences and messages timestamps. Therefore, TM-ToT can capture the changes of each topic over time. Subsequently, we aggregate all messages published by the same author to form a lengthy pseudo-document to tackle the data sparsity problem. Finally, we present a Gibbs sampling implementation for the inference of TM-ToT. We evaluate TM-ToT on a real dataset. In our experiments, TM-ToT outperforms Twitter-LDA by a large margin in terms of perplexity. Furthermore, the quality of the generated latent topics of TM-ToT is promising.

Keywords: Topic mining, microblog, Tibetan message, TM-ToT.

1 Introduction

Recent years we have witnessed an unprecedented growth of Weibo, which is a popular Chinese microblogging service that enables users to post and exchange short text messages (up to 140 characters). It was launched on 14 August 2009, and has more than 500 million registered users in 2012. Messages can be published through the website interface, SMS, or a wide range of apps for mobile

* Corresponding author.

devices. Therefore, Weibo facilitates real-time propagation of information. In particular, about 100 million messages are posted each day on Weibo. This makes it an ideal information network, which can tell people what they care about as it is happening in the society [1].

The whole Weibo data consist of multilingual texts, including the Tibetan messages, which are an essential part of this microblog platform. Such a vast amount of user-generated short messages in the Tibetan language implies a great opportunity for business providers, advertisers, social observers, data mining researchers, as well as governments. In this paper, our goal is to mine the latent topics from the Tibetan data, which plays a significant role in tracing the dynamics of the Tibetan community, contributing to uncover the public opinion of this community to the government.

Topic models can be a wise choice to discover latent topics from the large scale of document collections, such as scholarly journals and news articles. Most existing models are developed from the Latent Dirichlet Allocation (LDA) [2], whose basic idea is that each document is a finite mixture of topics and each topic is described by a distribution over words. The LDA-based models project each document into a low dimensional space where their latent semantic structure can be uncovered easily. Unfortunately, directly using conventional topic models to the Tibetan messages can result in unsatisfactory performance. The major reasons are two-fold. Firstly, the LDA-family models ignore the temporal information in the Tibetan messages. This contradicts the fact that the messages are composed of both plain texts and timestamps. In fact, assuming that each topic is a mixture distribution influenced by both word co-occurrences and timestamps sounds more reasonable in the microblog application. Secondly, the short messages cause severe data sparsity challenge, which makes the performance deteriorate significantly because the traditional topic models implicitly capture the document-level word co-occurrence patterns to reveal topics.

Considering the characteristics of the Tibetan messages (i.e., the rich temporal information and the severe data sparsity problem), we propose a novel model called TM-ToT, which is derived from ToT (Topic over Time) [3], aiming at mining latent topics effectively from the Tibetan messages. Firstly, we assume each topic is a mixture distribution influenced by both word co-occurrences and messages timestamps. Therefore, TM-ToT can capture the changes of each topic over time. Subsequently, we aggregate all messages published by the same author to form a lengthy pseudo-document to tackle the data sparsity problem. Finally, we present a Gibbs sampling implementation for the inference of TM-ToT. We evaluate TM-ToT on a real dataset. In our experiments, TM-ToT outperforms ToT by a large margin in terms of perplexity. Furthermore, the quality of the generated latent topics of TM-ToT is promising.

To sum up, the major contributions of our work are as follows:

- We study the problem of mining latent topics from the Tibetan messages to help government follow public opinion.
- We propose a novel topic model named TM-ToT and devise a Gibbs sampling for posterior inference.

- Extensive experiments on a real dataset are conducted. The results demonstrate the superiority of the proposed method.

The rest of this paper is organized as follows: the related work is discussed in Section 2; a novel topic model, TM-ToT, is proposed for topic mining from the Tibetan messages in Section 3; experimental results and discussions are presented in Section 4; finally we conclude our work in Section 5.

2 Related Work

Topic models are used to group words in a large-scale corpus into a set of relevant topics and have received increasing attention in the last decade. In this section, we briefly introduce the related work about topic models, from the basic models (i.e., LDA and ToT) to several complicated variants suitable for microblog applications.

LDA [2] is the most popular generative probabilistic model, which recognizes each document as a finite mixture over an underlying set of topics and describes each topic as a distribution of words that tend to co-occur. Through the elaborate probabilistic inference, LDA can successfully explore the hidden semantic structure of the corpus. One limitation of LDA is that it deems documents to be exchangeable during the topic modeling process. This assumption may not be tenable in some cases where the topics' occurrence and correlations change significantly over time, such as scholarly journals. ToT [3] relaxes the document exchangeable assumption by representing the mixture distribution over topics using both word co-occurrences and the text's timestamp. Experiments on real-world data sets demonstrate ToT can discover more salient topics associated with specific events and clearly localized in time.

The microblog services have gained increasing popularity and a large scale of user-generated content have been accumulated. Topic models seem appropriate to mine topics from textual documents with an unprecedented scale because of their principled mathematical foundation and effectiveness in exploratory content analysis. However, the conventional topic models usually fail to achieve satisfactory results when applied to the microblog data because the user-generated short messages cause the severe data sparsity challenge. To improve the performance of topic modeling for social media, researchers take the data sparsity into consideration and develop several extensions of LDA. Ramage et al. [4] proposed a scalable implementation of a partially supervised learning model called Labeled LDA to characterize users and messages. Cha et al. [5] incorporated popularity in topic models for social network analysis. The authors argued that a popular user has very important meaning in the microblog dataset and should be carefully handled to refine the probabilistic topic models. Zhao et al. [6] develops a user-based aggregation method, Twitter-LDA, to integrate the tweets published by individual user into a lengthy pseudo-document before training LDA. Zhang et al. [7] introduced a novel probabilistic generative model MicroBlog-Latent Dirichlet Allocation (MB-LDA) for large scale microblog mining. The MB-LDA utilizes both contactor relevance relation and document relevance relation to

improve the topic mining result. Tang et al. [8] pointed out that classical topic models will suffer from significant problems of data sparseness when applied to social media. Resorting to other types of information beyond word co-occurrences at the document level can significantly improve the performance of topic modeling. Therefore, the authors proposed a general solution that is able to exploit multiple types of contexts without arbitrary manipulation of the structure of classical topic models. Diao et al. [9] claimed that users on microblogs often talk about their daily lives and personal interests besides talking about global popular events. Therefore, they proposed a novel topic model that considers both temporal information of messages and users' personal interests. The model assumes that each message only consists of a single topic rather than a mixture of topics.

However, none of the topic models mentioned above is specifically tailored to the Tibetan messages, which is an important part of user-generated content in Weibo platform. for social network analysis, such as government follow public opinion. We thus propose a novel topic model named TM-ToT to extract interesting hidden topics from the Tibetan messages, by taking both temporal and context information into consideration.

3 TM-ToT Model

In this section, we introduce a novel generative probabilistic model TM-ToT for topic mining from Tibetan messages. We first describe the framework of TM-ToT, and then design a Gibbs sampling to infer our model. The notations used in TM-ToT are summarized in Table 1.

Table 1. Notations Used in TM-ToT

SYMBOL	DESCRIPTION
D, U, K, V	number of messages, users, topics, and unique words, respectively
N_d	number of words in message d
n_k^v	number of words v are assigned to topic k
n_u^k	number of words associated with user u
θ_u	the multinomial distribution of topics to user u
α, β	Dirichlet priors for θ_u and ϕ_k , respectively
ϕ_k	the multinomial distribution of words to topic k
ψ_k	the beta distribution of timestamps to topic k
$w_{d,i}$	i th word in message d
$z_{d,i}$	topic of i th word in message d
$z_{-(d,i)}$	topic assignments for all words except $w_{d,i}$
$w_{-(d,i)}$	all words except $w_{d,i}$ in message d
$t_{d,i}$	the timestamp associated with i th word in message d
\bar{t}_k, s_k^2	the sample mean and variance of timestamps belonging to topic k , respectively

3.1 TM-ToT Framework

Topic models achieve promising performance when documents present sufficient and meaningful signals of word co-occurrences. However, they fail to perform effectively when applied to user-generated short messages where the word co-occurrences are limited and noisy. Therefore, we should resort to rich context information (e.g., time and authorship) beyond word co-occurrences within plain texts to improve the quality of topic modeling in social media. Motivated by this intuition, the Tibetan messages are grouped into different subsets by their authors (an author refers to the user who posts a message) and each subset can be seemed as a pseudo-document, whose topic distribution inherently reflects the user’s intrinsic interests. To utilize the temporal information, we assume each topic as a mixture distribution influenced by both word co-occurrences and timestamps of the Tibetan messages.

According to the above analysis, we devise a novel topic model named TM-ToT to deal with the topic modeling task for the Tibetan messages. Our proposed method firstly assembles messages written by the same user as a pseudo-document to alleviate the data sparsity problem. Then, TM-ToT utilizes the temporal information during the topic modeling process in the same way as ToT does, making it possible to create a topic with a broad time distribution and draw a distinction between topics due to their changes over time. In TM-ToT, the beta distribution seems to be an appropriate choice to describe various skewed shapes of rising and falling topic prominence in social media. The Bayesian graphical framework of TM-ToT is illustrated in Figure 1.

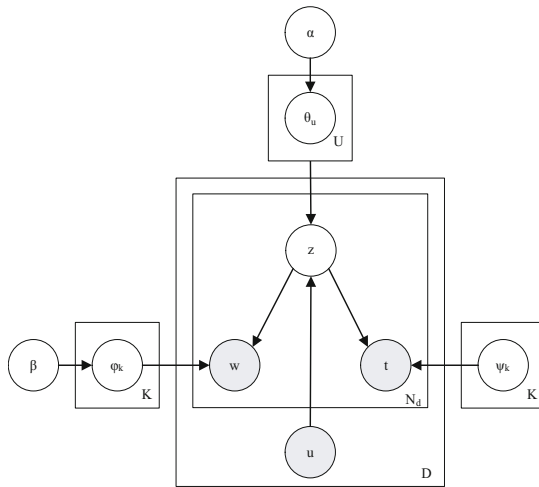


Fig. 1. Bayesian Graphical Framework of TM-ToT

Let θ_d denote the topic distribution of message d , then the detailed generative process of TM-ToT is as follows:

1. For each topic $k \in [1, K]$:
 - (a) Draw a multinomial ϕ_k from a Dirichlet prior β ;
2. For each message $d \in [1, D]$, published by user u :
 - (a) Draw a multinomial θ_u from a Dirichlet prior α ;
 - (b) Assign the value of θ_u to θ_d ;
 - (c) For each word $i \in [1, N_d]$:
 - i. Draw a topic $z_{d,i}$ from the multinomial θ_d ;
 - ii. Draw a word $w_{d,i}$ from the multinomial $\phi_{z_{d,i}}$;
 - iii. Draw a timestamp $t_{d,i}$ from the beta $\psi_{z_{d,i}}$;

As shown in the above process, for each message d , its posterior distribution of topics θ_d depends on the authorship information.

$$P(\theta_d|\alpha) = P(\theta_u|\alpha). \tag{1}$$

The joint probability distribution of message d is:

$$P(\mathbf{w}, \mathbf{t}, \mathbf{z}|\alpha, \beta, \Psi) = P(\mathbf{w}|\mathbf{z}, \beta)P(\mathbf{t}|\mathbf{z}, \Psi)P(\mathbf{z}|\alpha). \tag{2}$$

To sum up, the formal description of the generative process in TM-ToT is:

$$\begin{aligned} \theta_d &= \theta_u|\alpha \sim \text{Dirichlet}(\alpha) \\ \phi_k|\beta &\sim \text{Dirichlet}(\beta) \\ z_{d,i}|\theta_d &\sim \text{Multinomial}(\theta_d) \\ w_{d,i}|\phi_{z_{d,i}} &\sim \text{Multinomial}(\phi_{z_{d,i}}) \\ t_{d,i}|\psi_{z_{d,i}} &\sim \text{Beta}(\psi_{z_{d,i}}) \end{aligned}$$

3.2 TM-ToT Inference

The key issue for generative probabilistic models is to infer the hidden variables by computing their posterior distribution given the observed variables. As show in Figure 1, the temporal metadata, words and users are observed variables, while the topic structure and its changes over time are hidden variables.

The inference can not be done exactly in TM-ToT. We employ Gibbs sampling to perform approximate inference due to its speediness and effectiveness. In the Gibbs sampling procedure, we need to calculate the conditional distribution $P(z_{d,i}|\mathbf{w}, \mathbf{t}, z_{-(d,i)}, \alpha, \beta, \Psi)$. Taking advantage of conjugate priors, the joint distribution $P(\mathbf{w}, \mathbf{t}, \mathbf{z}|\alpha, \beta, \Psi)$ can be resolved into several components:

$$P(\mathbf{w}|\mathbf{z}, \beta) = \left(\frac{\Gamma(\sum_{v=1}^V \beta_v)}{\prod_{v=1}^V \Gamma(\beta_v)} \right)^K \prod_{k=1}^K \frac{\prod_{v=1}^V \Gamma(n_k^v + \beta_v)}{\Gamma(\sum_{v=1}^V (n_k^v + \beta_v))} \tag{3}$$

$$P(\mathbf{t}|\mathbf{z}, \Psi) = \prod_{d=1}^D \prod_{i=1}^{N_d} P(t_{d,i}|\psi_{z_{d,i}}) \tag{4}$$

$$P(\mathbf{z}|\alpha) = \left(\frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \right)^U \prod_{u=1}^U \frac{\prod_{k=1}^K \Gamma(n_u^k + \alpha_k)}{\Gamma(\sum_{k=1}^K (n_u^k + \alpha_k))} \tag{5}$$

We can conveniently obtain the conditional probability by using the chain rule.

$$\begin{aligned} & P(z_{d,i}|\mathbf{w}, \mathbf{t}, z_{-(d,i)}, \alpha, \beta, \Psi) \\ &= P(z_{d,i}|\mathbf{w}, \mathbf{t}, z_{-(d,i)}, \alpha, \beta, \Psi) \\ &\propto \frac{n_{z_{d,i}}^{w_{d,i}} + \beta_{w_{d,i}} - 1}{\sum_{v=1}^V (n_{z_{d,i}}^v + \beta_v) - 1} \times (n_{z_{d,i}}^v + \alpha_{z_{d,i}} - 1) \times p(t_{d,i}|\psi_{z_{d,i}}) \end{aligned} \quad (6)$$

We sample the posterior distribution using Gibbs sampling until it reaches a convergence for all messages. Then, we obtain the multinomial parameters as follows:

$$\phi_{k,v} = \frac{n_k^v + \beta_v}{\sum_{v=1}^V (n_k^v + \beta_v)} \quad (7)$$

$$\theta_{u,k} = \frac{n_u^k + \alpha_k}{\sum_{k=1}^K (n_u^k + \alpha_k)} \quad (8)$$

For the sake of simplicity and speed, Ψ is updated after each Gibbs sample by the method of moments estimates:

$$\hat{\psi}_{k,1} = \bar{t}_k \left(\frac{\bar{t}_k(1 - \bar{t}_k)}{s_k^2} - 1 \right) \quad (9)$$

$$\hat{\psi}_{k,2} = (1 - \bar{t}_k) \left(\frac{\bar{t}_k(1 - \bar{t}_k)}{s_k^2} - 1 \right) \quad (10)$$

After finishing the inference process, TM-ToT can detect topics from messages and assign the most representative words to each topic. Additionally, TM-ToT can detect the changes of each topic over time by a beta distribution with parameters from Equation 9 and 10. In summary, TM-ToT is a convenient tool for topic mining from the Tibetan messages.

4 Experiments

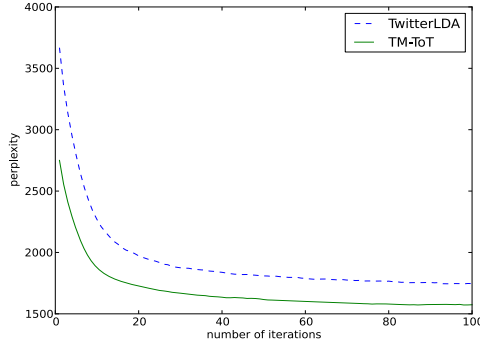
In this section, TM-ToT is evaluated empirically over a crawl of Tibetan messages from four different perspectives: the perplexity of held-out content, the quality of the generated latent topics and the dynamic topics analysis.

4.1 Dataset

To validate TM-ToT, we use a Weibo dataset with 70,768 messages from January 2013 to October 2013. Messages are usually short and inaccurate uses of language, which makes the quality of messages varies a lot from each other, therefore specific data preprocessing techniques are required to filter low-quality messages. We firstly use a novel Tibetan word segmentation to divide the original message text into meaningful units. Secondly, we prepare a Tibetan stop word list in advance to remove Tibetan stop words in original messages, since these frequent words do not have much meaning. Thirdly, we filter out words with less

Table 2. Description of Dataset

# of messages	30,260
# of unique words	8,987
# of users	182
# of tokens in messages	411,694
average length of each message	13
minimal timestamp (seconds)	1356969600.0
maximal timestamp (seconds)	1381248000.0


Fig. 2. Perplexities of Different Methods

than 10 occurrences in our dataset and only keep the messages with more than 8 terms. Finally, we build a medium dataset containing 30,260 messages collected from 182 selected users for experiment evaluations. The detail information of our dataset is shown in Table 2. The simulations are carried out on an Intel Dual Core PC with 2.67 GHz CPU and 2 GB RAM.

4.2 Perplexity of Held-out Content

The metric perplexity is a widely used method to measure the performance of a topic model, which indicates the uncertainty in predicting a single word. A lower perplexity indicates better performance. To compute the perplexity of all messages, we use the formula as below:

$$Perplexity(D) = \exp \left(- \frac{\sum_d \sum_i^{N_d} \log p(w_{d,i})}{\sum_d N_d} \right) \quad (11)$$

Twitter-LDA [6] is chosen for comparison. For the sake of fairness, the parameters α , and β in both models are set to 0.1 and 0.01, respectively. Figure 2 shows the perplexities for our TM-ToT and the baseline with 50 latent topics until they reach the convergence after enough iterations. We observe that TM-ToT achieves the best perplexity. This means that integrating the temporal

Topic 1		Topic 2		Topic 3		Topic 4	
Tibetan	English	Tibetan	English	Tibetan	English	Tibetan	English
མཚན་མོག་	outstanding	ཏུས་	time	འཕྲིན་ཚུ་	different	མིན་	not
ལྷ་མ་	Living Buddha	ཚེ་	living	ལྷོད་	politics	བླ་ན་པ་	religion
ཚེས་	thing	ཏུས་རབས་	era	མིན་	not	རྗེ་	everyone
ཐམས་ཅད་	everything	ཕན་ཚུན་	mutual	འདོད་	willing	རིག་གནས་	knowledge
ལྗང་	change	ཐོབ་	obtain	འདོད་པ་	hobbit	པར་ཤོག་	paper
ཚེ་	living	མཚོན་	decorate	རིགས་	category	པར་ཤོག་	wise man
བར་	interval	མུར་	unable	རིལ་	class	བཀའ་འདིན་	kindness
མཛད་	do	གཤན་	permanence	ཡོད་ཚད་	all	རྒྱུ་	self
ཏུས་རབས་	era	དྲོགན་མཚན་	precious	མཐུན་རྐྱེས་	unite	ལྷ་མ་	mark
རྩ་བ་	fundamental	རྒྱལ་ལྡན་	tradition	ནང་	inside	ལྷོ་	lotus

Fig. 3. Top 10 Words for Latent Topics (K=50)

information into the topic modeling process leads to better performance. Note that the perplexities of both models do not change significantly when the number of iterations is greater than 40.

4.3 Effectiveness of Latent Topics

The main purpose of topic models for messages is to find out interesting topics from the overwhelming information. One typical method of judging the effectiveness of topic models is to print words with top weights for the latent topics and judge them by experience [10]. Figure 3 shows the quality of latent topics generated by our model. There are four topics listed out of total 50 topics, each of which is represented with the top 10 words due to the limit of space. We can learn that Topic 1 is about “Living Buddha”; Topic 2 is about “Eternal Time”; Topic 3 is about “Politics”; Topic 4 is about “Religion”. The key words of each topic are accurate enough to recognize and these topics are pretty independent with each other.

4.4 Dynamic Topics Analysis

The ability of modeling the changes of topics over time is very important for topic mining in microblogs. TM-ToT combines the temporal information to capture the dynamic topics. Figure 3 illustrates all beta distributions of each topic over time when the number of topics is set to 50. An immediate and obvious effect of this is to understand more precisely when and how long the topical

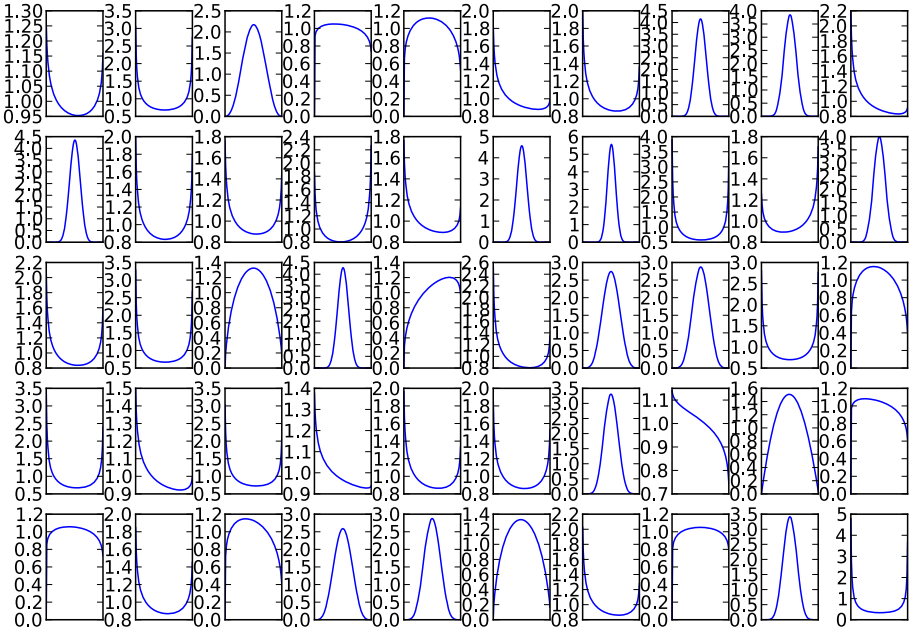


Fig. 4. Beta Distributions of Topics over Time (K=50). Note that, the X-axis is from 0 to 1.

trend was occurring. Although the beta distribution is adopted for representing various skewed schemes of rising and falling topic prominence, there still exist several salient types. For example, the uniform distribution is common in our experiment. Note that, even the similar shape of distributions have different means and variances. Thus, the changes of topics over time can be distinguished easily.

5 Conclusions

In this paper, we present and evaluate a time-aware topic model TM-ToT mixed with user’s intrinsic interests, for effectively modeling and analyzing the topics that naturally arise in Tibetan microblogs. TM-ToT is able to capture the changes in the occurrence of topics by assuming that each topic is a mixture distribution influenced by both word co-occurrences and timestamps of microblogs. Moreover, the author relationship information is used to solve the severe data sparsity problem. Finally, the inference of TM-ToT is completed by a Gibbs sampling. Extensive experiments on a real dataset demonstrate that TM-ToT outperforms its competitor.

In the future work, we will focus on investigating more social network information, such as follow/following relations and URLs, to improve the performance of topic models. How to describe the temporal metadata distributions is another

interesting direction. Finally, we will devise more elaborate and effective model to merge the social network information.

Acknowledgments. This work is supported by the National Social Science Fund of China (13BXW037) and the Ministry of Education Chunhui Project (Z2011023).

References

1. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: A content-based approach to geo-locating twitter users. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 759–768 (2010)
2. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022 (2003)
3. Wang, X., McCallum, A.: Topics over time: A non-markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 424–433 (2006)
4. Ramage, D., Dumais, S., Liebling, D.: Characterizing microblogs with topic models. In: Proceedings of the International AAAI Conference on Weblogs and Social Media, pp. 130–137 (2010)
5. Cha, Y., Bi, B., Hsieh, C.-C., Cho, J.: Incorporating popularity in topic models for social network analysis. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 223–232 (2013)
6. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: Proceedings of the European Conference on Information Retrieval, pp. 338–349 (2011)
7. Zhang, C., Sun, J.: Large scale microblog mining using distributed mb-lda. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 1035–1042 (2012)
8. Tang, J., Zhang, M., Mei, Q.Z.: One theme in all views: Modeling consensus topics in multiple contexts. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 5–13 (2013)
9. Diao, Q., Jiang, J., Zhu, F., Lim, E.: Finding bursty topics from microblogs. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 536–544 (2012)
10. Xu, Z., Zhang, Y., Wu, Y., Yang, Q.: Modeling user posting behavior on social media. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 545–554 (2012)

Chinese Microblog Entity Linking System Combining Wikipedia and Search Engine Retrieval Results

Zeyu Meng, Dong Yu, and Endong Xun

Inter. R&D center for Chinese Education,
Beijing Language and Culture University, 100083, Beijing, China
mengzeyu_blcu@163.com, {yudong, edxun}@blcu.edu.cn

Abstract. Microblog has provided a convenient and instant platform for information publication and acquisition. Microblog's short, noisy, real-time features make Chinese Microblog entity linking task a new challenge. In this paper, we investigate the linking approach and introduce the implementation of a Chinese Microblog Entity Linking (CMEL) System. In particular, we first build synonym dictionary and process the special identifier. Then we generate candidate set combining Wikipedia and search engine retrieval results. Finally, we adopt improved VSM to get textual similarity for entity disambiguation. The accuracy of CMEL system is 84.35%, which ranks the second place in NLPCC 2014 Evaluation Entity Linking Task.

1 Introduction

In recent years, microblog has provided a convenient and instant platform for information publication and acquisition. Bridging microblog post with knowledge bases (KB) can facilitate kinds of tasks such as news detection, information aggregation and correlation recommendation. Chinese Entity Linking Task is aimed to link mentions in microblog-genre text with the corresponding entities in knowledge base, or return "NIL" if the entity is out of knowledge base. Unlike formal text such as news or papers, microblog post is short (no more than 140 characters), noisy (abbreviated form, typos, network-format words etc.) and real-time (new words occurs, words popularity changes), which make Chinese microblog entity linking task a new challenge.

There are two main issues in entity linking task: mention ambiguity and mention variation. The mention ambiguity refers to polysemy phenomenon of nature language: one mention is potentially related to many different KB entries. The mention variation means that a named entity may be expressed in different ways including nickname, previous name, abbreviation or sometimes misspellings.

In this paper, we adopt a cascade approach to identify links between mentions in microblog and entities in knowledge base. We first construct synonym dictionary to deal with mention variation problem. Then, we process special identifier to identify whether the username after "@" is celebrity or ordinary people. After that we generate candidate set combining Wikipedia and search engine retrieval results. Finally, we adopt improved VSM to get textual similarity for entity disambiguation.

2 Related Work

One of the classical methods is to extract discriminative features of a mention from its context and an entity from its description, then link a mention to the entity which is most similar with it.[1-3] Zheng et al., Zhang et al. and Zhou et al. adopted learning to rank techniques which utilizing relations among candidate entities.[4-6] Those context similarity based methods heavily rely on features but fail in incorporate heterogeneous entity knowledge.

There are also some inter-dependency based entity linking methods. Those methods assumed that the entities in the same document are related to each other and the referent entity is the entity which is most related to its contextual entities.[7-9] The inter-dependency based methods are usually designed for long and normative documents, but do not suit well for microblog genre text from which few feature or related entities can be extracted due to its short and informal content.

Recently, more and more works have been focusing on short informal text. Guo et al. [10] proposed a context-expansion-based and a graph-based method for microblog entity linking by leveraging extra posts. In NLPCC 2013 evaluation task, Miao et al. [11] introduced a microblog semantic annotation system which includes mention expansion by knowledge repository and entity disambiguation considering lexical matching, popularity probability and textual similarity. Zhu et al. [12] used improved pinyin edit distance, suffix vocabulary matching and entity clustering disambiguation. These methods perform well in microblog genre text but do not fully utilize web sources such as search engine retrieval results and information provided by Weibo platform.

3 The Approach

3.1 Overview of the Framework

The main process of Chinese Microblog Entity Linking system contains 3 modules: preprocessing, candidate generation and entity disambiguation. The preprocessing module includes normalizing and indexing the knowledge base, building the dictionary of synonyms and processing special identifier. Candidate generation module uses voting mechanism to combine the retrieval result of Wikipedia and search engine. At last, entity disambiguation module generates the optimal result adopting improved vector space model. Figure 1 shows the framework of the system.

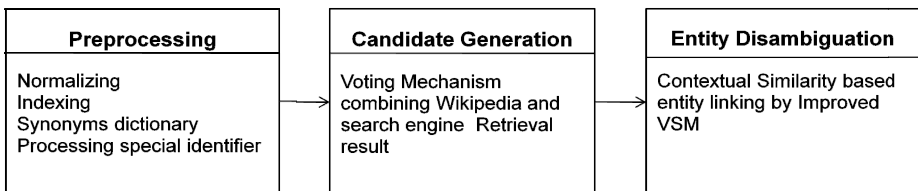


Fig. 1. The framework of Chinese Microblog Entity Linking system

The preprocess module makes preparation for the next two steps. Especially, if there is special identifier in microblog to indicate usernames, we get the user’s popularity information from Weibo, thus resolving the problem that common user has the same name with celebrity.

The candidate generation module considers both Wikipedia and search engine retrieval results. In Wikipedia, redirect pages and disambiguation pages could be recognized if a mention is ambiguous. Search engine retrieval result integrates popularity information, and provides error correction function to recall potential candidate.

The entity disambiguation module introduces improved vector space model to avoid word segmentation error which may affect the accuracy of similarity calculation in VSM.

In Microblog Entity Linking task, given a mention, M , the system is a function $F(M)$, which links M to its referent entity in KB, or NIL if outside KB. Figure 2 shows workflow of CMEL system:

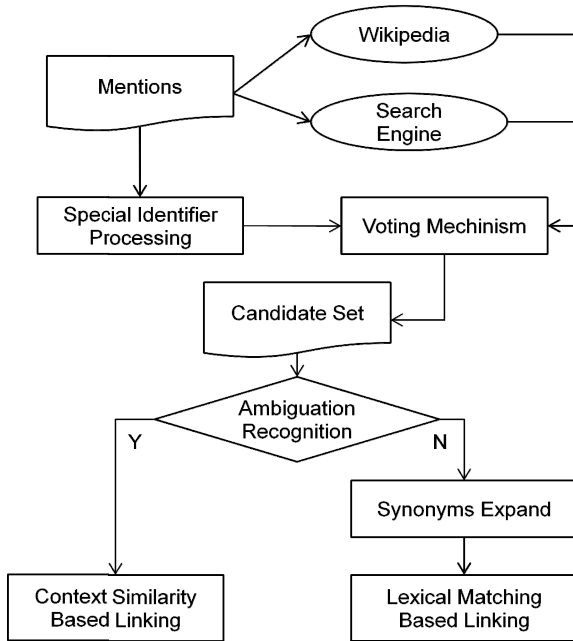


Fig. 2. The workflow of Chinese Microblog Entity Linking system

The implementation details of each step will be described in next 3 section.

3.2 Data Preprocess

The preprocess module makes preparation for next modules, including normalizing and indexing the knowledge base, building the dictionary of synonyms and processing special identifier.

Normalizing and Indexing Knowledge Base

Before we index knowledge base, we have to normalize the knowledge base first. Since the tag name contains both Chinese and English, we use English-Chinese dictionary to normalize all tag names as Chinese. We also normalize punctuations from underline to whitespace and generate a paragraph of description text from original xml format. For word segmentation and stop words filtering job, we adopt SmartChineseAnalyzer carried in Apache Lucene 4.6¹. Table 1 shows original text in KB with its normalized text.

Table 1. The original knowledge base vs. preprocessed knowledge base

Original knowledge base	Preprocessed knowledge base
<entity enity_id="WKB136" title="自由软件基金会">	[ID]WKB136 [NAME]自由软件基金会
<leader_name>理查德·斯托曼</leader_name>	领导者 姓名 理 查 德 斯 托 曼
<motto>free software, free society</motto>	座右铭 free softwar free societi

Then, we index entities with corresponding normalized description paragraph for KB retrieval later.

Building Synonyms Dictionary

Words in microblog are flexible: nickname, previous name and abbreviation often occurred and the statement style is informal containing large amount of network words. Building a synonyms dictionary is an effective way to deal with mention variation issue. We extract those information whose tag name is “简称”, “本名”, “绰号” etc. from knowledge base first. For instance, we get “皖” as alias of “安徽省” and “李珍基” as previous name of “温流”. Also, redirect pages in Wikipedia give us a good indicator for synonyms. For example, for query “老蒋”, Wikipedia redirect it to the article “蒋中正”. Another resource to build synonyms dictionary is from first paragraph of each article in Wikipedia. We adopt Miao’s methods [16] to extract all those information to build the synonyms dictionary.

Processing Special Identifier

Some mentions in microblog are, or part of, usernames after the special identifier “@”. For example, “@李巍LW1982”, obviously, the mention “李巍” is just a common name, not referring to the football player “李巍”; while in “与中国时尚界最具潜力和最受瞩目的新晋超模 @李丹妮”, the mention “李丹妮” refers to the famous model “李丹妮”, indeed. To deal with this kind of mentions, we extract the username’s amount of fans from Weibo platform. The username “李巍LW1982” only has 52 fans while the username “李丹妮” has more than 380 thousand fans. So we set

¹ <http://lucene.apache.org/>

50 thousand of fans as the threshold to identifying the celebrity’s name with ordinary user name.

3.3 Candidate Generation by Voting Mechanism

Since the knowledge base in this task is from Wikipedia, for each mention, we retrieve it in Wikipedia first. However, Wikipedia’s searching mechanism is naive without error correction or popularity knowledge: Its redirect page may be incorrect, and sometimes it cannot return any result for misspelling case. For instance, when search the mention “李小露”, Wikipedia cannot return any matched result, but actually it is the misspelling form of mention “李小璐” which is supposed to be linked with the famous actress in China. Therefore, we import retrieval result of the largest Chinese search engine, Baidu, into our candidate generation module. Baidu provides error correction function and takes more factors such as popularity reference into account. For the case mentioned above, Baidu can correct the misspelling “李小露” as “李小璐” and return the desired retrieval result.

In this module, we introduce voting mechanism to combine the retrieval results of Wikipedia and Baidu search engine. Through the voting process, each retrieval result will be given a vote score, constituting candidate set of a mention.

Wikipedia Retrieval Result

For a mention, the Wikipedia’s returned result will be divided into three types: 1) not existing, 2) only one existing result (may be redirected), 3) disambiguation page. For case 1, we mark Wikipedia retrieval result as NIL with score 0.9; for case 2, we give the result 1.0 score; for case 3, we extract all entities in the disambiguation page and give each of them score of 0.5. Thus, for each mention, M , if it is linked to candidate C , the vote score by Wikipedia is represented as follows,

$$V_{\text{Wiki}}(C|M) = \begin{cases} 0.9, & C = \text{NIL} \\ 1.0, & C = \text{the retrieval result} \\ 0.5, & C \in \text{entities set in disambiguation page} \end{cases} \quad (1)$$

Baidu Search Engine Retrieval Results

We submit mentions with assist query “维基百科” and “中文维基百科” to Baidu API. If top 1 retrieval result is linked to the website “zh.wikipedia.org/wiki/...”, we extract the entity name in the title of returned result, regard it as a candidate, C , with score 0.5; while if the top 1 is linked to other website, we regard the mention, M , as an outside KB query, and set score 0.4 to be “NIL”. Then, with Baidu search engine, we get 2 votes score expressed as:

$$V_{\text{Baidu}}(C|M) = \begin{cases} 0.5, & C = \text{entity name in top 1 result} \\ 0.4, & C = \text{NIL} \end{cases} \quad (2)$$

Final Vote Score

Now that given a mention M , we have 1 vote of Wikipedia and 2 votes of Baidu search engine, for each candidate $C_i \in \text{Set}_{\text{Wiki}} \cup \text{Set}_{\text{Baidu}}$, the final vote score will be:

$$V_{\text{final}}(C_i | M) = \alpha V_{\text{Wiki}}(C_i | M) + \beta V_{\text{Baidu1}}(C_i | M) + \beta V_{\text{Baidu2}}(C_i | M) \quad (3)$$

Testing on the training data set, we assign $\alpha = 1.0, \beta = 0.5$.

To generate final candidate set, if a mention is unambiguous, we choose C_i which has the maximum V_{final} as final candidate, expand it with synonyms dictionary, and search in the knowledge base to get the matched entity. If a mention is ambiguous, we add all the entities in the disambiguation page to candidate set with corresponding vote score, V_{final} .

3.4 Entity Disambiguation Based on Improved VSM

As is mentioned above, after candidate generation module, we get candidate set of the ambiguous mention. Since candidate set contains both the potential named entity and its vote score generated during voting mechanism, we adopt vector space model to get textual similarity between microblog text and content of candidate in knowledge base. The context of mention M and candidate C_i will be expressed as vector $(\langle t_1, w_1 \rangle, \langle t_2, w_2 \rangle, \dots, \langle t_i, w_i \rangle, \dots, \langle t_n, w_n \rangle)$.

However, in preprocess module, word segmentation error is inevitable. To avoid error in word segmentation affecting the accuracy of similarity in VSM, we take the length of t_i into account and the weight, w_i of each t_i will be

$$w_i = \text{tfidf}(t_i) \cdot \text{length}(t_i) \quad (4)$$

We use cosine distance to measure the similarity. So, in improved VSM, similarity between M and C_i is measured as:

$$\text{Sim}(C_i, M) = \frac{C_i \cdot M}{\|C_i\| \cdot \|M\|} = \frac{\sum_{j=1}^n (w_{ij} \cdot w_{Mj})}{\sqrt{\sum_{j=1}^n w_{ij}^2} \cdot \sqrt{\sum_{j=1}^n w_{Mj}^2}} \quad (5)$$

Finally, the mapping function, $F(M)$, from a mention to its referent entity is expressed as follow:

$$F(M) = \begin{cases} C_i, & \text{if } \text{argmax}(\text{Sim}(C_i, M) + V_{\text{final}}(C_i | M)) \geq \text{length}(C_i) \\ \text{NIL}, & \text{else} \end{cases} \quad (6)$$

4 Experiments

4.1 Experimental Setup

In the experiment, we use the data set published by Natural Language Processing and Chinese Computing Conference (NLPC) 2014 for Chinese Entity Linking evaluation task. The reference knowledge base used in this task is built from the InfoBoxes

of the Chinese part of Wikipedia dumps in 2013, which contains 400,000 entities with all kinds of tags of properties. The training data set contains 169 microblog post with 250 mentions, and the test data set contains 570 microblog post with 607 mentions. In addition, there exist 2 mentions in training set which are usernames after special identifier “@” and 32 in test data set, showed in Table 2.

Table 2. The statistics of the annotated results

Data set	Microblog posts	Mentions	Special identifier “@”
Training data set	169	250	2
Test data set	570	607	32

4.2 Experimental Results

Table 3 shows the contribution that special identifier “@” makes. By processing special identifier, the system’s overall accuracy is improved from 80.07% to 84.35%, which indicates this is an effective method to differentiate celebrity name from common user name.

Table 3. The performance of processing special identifier “@”

System	Accuracy
System without processing “@”	0.8007
System with processing “@”	0.8435

Table 4 reports the performance of our Chinese Microblog Entity Linking (CMEL) system compared with the number 1 system in EL evaluation task. For evaluation, we use the overall micro-averaged accuracy, and further compute the precision, recall, F-1 measures over in-KB entities and NIL entities, respectively.

Table 4. The EL evaluation results

System	Overall		in-KB		NIL		
	Accuracy	Precision	Recall	F1	Precision	Recall	F1
#1	0.8682	0.8078	0.8598	0.8330	0.9202	0.8746	0.8969
CMEL	0.8435	0.8103	0.7765	0.7930	0.8672	0.8950	0.8809

The overall accuracy of our CMEL system is 84.35%, which ranks the second place in Entity Linking evaluation task. From table 4, we can see that, the precision of in-KB result and recall of NIL result are higher than #1’s, while the recall of in-KB result and precision of NIL are not that satisfying. It means our system is such strict in candidate generation process that linking some uncertain mention to NIL incorrectly. Generally speaking, combining Wikipedia and search engine can achieve promising result in real world dataset.

5 Conclusion

In this paper we introduce how our Microblog Entity Linking system works to link mentions in microblog posts with named entity in knowledge base. We combine Wikipedia and search engine retrieval results to generate candidate set and adopt improved VSM in entity disambiguation step. In addition, we process the special identifier “@” to differentiate celebrity name from common user name. Experiment result on real world microblog datasets shows this entity linking system is promising and effective. In future work, we will focus on colloquial and casual address of personal name according to its context. About entity disambiguation, finding the strategy to accurately and properly expand microblog-genre text with inter-dependency will be main part of our future work.

References

1. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 708–716 (2007)
2. Mihalcea, R., Csomai, A.: Wikify! Linking Documents to Encyclopedic Knowledge. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 233–242 (2007)
3. McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D., Dreyer, M.: HLTCOE Approaches to Knowledge Base Population at TAC 2009. In: Proceedings of Text Analysis Conference (TAC) (2009)
4. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to Link Entities with Knowledge Base. In: The Proceedings of the Annual Conference of the North American Chapter of the ACL, pp. 483–491 (2010)
5. Zhang, W., Su, J., Tan, C.L., Wang, W.T.: Entity Linking Leveraging Automatically Generated Annotation. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1290–1298 (2010)
6. Zhou, Y., Nie, L., Rouhani-Kalleh, O., Vasile, F., Gaffney, S.: Resolving Surface Forms to Wikipedia Topics. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1335–1343 (2010)
7. Han, X.P., Sun, L.: A Generative Entity-Mention Model for Linking Entities with Knowledge Base. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 945–954 (2011)
8. Milne, D., Witten, I.H.: Learning to Link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518 (2008)
9. Chen, Z., Ji, H.: Collaborative Ranking: A Case Study on Entity Linking. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 771–781 (2011)
10. Guo, Y., Qin, B., Liu, T., Li, S.: Microblog Entity Linking by Leveraging Extra Posts. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 863–868 (2013)
11. Miao, Q., Lu, H., Zhang, S., Meng, Y.: Simple Yet Effective Method for Entity Linking in Microblog-Genre Text. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) NLPCC 2013. CCIS, vol. 400, pp. 440–447. Springer, Heidelberg (2013)
12. Zhu, M., Jia, Z., Zuo, L., Wu, A., et al.: Research on Entity Linking of Chinese Micro Blog. *Journal of Peking University (Natural Science Edition)* 01, 73–78 (2014) (in Chinese)

Emotion Cause Detection with Linguistic Construction in Chinese Weibo Text

Lin Gui¹, Li Yuan¹, Ruifeng Xu^{1*}, Bin Liu¹, Qin Lu², and Yu Zhou¹

¹ Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
{guilin.nlp,yuanlisail}@gmail.com, xuruifeng@hitsz.edu.cn, bliu@insun.hit.edu.cn, zhouyu.nlp@gmail.com

² Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
csluqin@comp.polyu.edu.hk

Abstract. To identify the cause of emotion is a new challenge for researchers in nature language processing. Currently, there is no existing works on emotion cause detection from Chinese micro-blogging (Weibo) text. In this study, an emotion cause annotated corpus is firstly designed and developed through annotating the emotion cause expressions in Chinese Weibo Text. Up to now, an emotion cause annotated corpus which consists of the annotations for 1,333 Chinese Weibo is constructed. Based on the observations on this corpus, the characteristics of emotion cause expression are identified. Accordingly, a rule-based emotion cause detection method is developed which uses 25 manually complied rules. Furthermore, two machine learning based cause detection methods are developed including a classification-based method using support vector machines and a sequence labeling based method using conditional random fields model. It is the largest available resources in this research area. The experimental results show that the rule-based method achieves 68.30% accuracy rate. Furthermore, the method based on conditional random fields model achieved 77.57% accuracy which is 37.45% higher than the reference baseline method. These results show the effectiveness of our proposed emotion cause detection method.

Keywords: Emotion cause detection, corpus construction, Chinese Weibo.

1 Introduction

The rise of social media has produced flooded of text such as Blogs and micro-blogging. How to analyze the emotions from these texts is becoming a new challenge for natural language processing researchers. Generally speaking, there are three basic tasks in emotion computation. 1. Emotion analysis, which focuses on how to classify the emotion categories of the texts and extract the holder/target of the emotion; 2. Emotion prediction, which predicts the readers' emotion after they read the given texts; and 3. Emotion cause detection, which extracts the cause of emotion in the text.

* Corresponding author.

Currently, most existing works on emotion computing focus on the emotion analysis [1-5] and emotion prediction [6-9]. Few works on emotion cause detection are reported. Meanwhile, the majority of these works follows linguistic based approach [10-12]. As we know, emotion is “physiological arousal, expressive behaviors and conscious experience” [13,14]. This motivated the psychological based emotion cause detection approach which emphasizes the “physiological arousal” rather than “empirical cue” in linguistic based approach..

The lack of emotion cause annotated corpus is a big barrier to emotion cause detection research. Especially, based on our knowledge, there is no open emotion cause annotated corpus on Chinese Weibo text. Therefore, in the first part of this study, an emotion cause annotated corpus is designed and developed. In order to reveal the relationship between emotion expression and emotion cause from the point of view of psychology, the framework which annotates the expression related to emotion cause is designed. Following this framework, the emotion cause annotated corpus corresponding to 1,333 Weibo is constructed.

Based on the observation on this corpus, three emotion cause detection methods are investigated. The first one is a rule based method. 25 rules based on the syntactic and semantic characteristics related to emotion cause expression are manually compiled for emotion cause detection. Furthermore, two machine learning based methods are developed. One is based on classification method using support vector machines (SVMs) model and the other one is based on sequence labeling based method using conditional random fields (CRFs) model. They use the same feature space. The experimental results show that the CRFs based method achieves the best accuracy of 77.57% which improves from the reference baseline method for 37.45%. This improvement shows the effectiveness of our proposed emotion cause detection method based on CRFs model.

The rest of this paper is organized as follow: section 2 will introduce the annotation format of our corpus; section 3 is our proposed methods for emotion cause extraction and section 4 is the experiment; section 5 will make a conclusion of this paper.

2 Construction of Emotion Cause Annotated Corpus

In this study, the corpus adopted in NLP&CC 2013 emotion analysis share task (in short NLPCC13 dataset) is selected as the basic annotation resource. NLPCC13 dataset annotates up to two basic emotion categories to each sentence and Weibo. In this dataset, 7 basic emotion categories, namely, *fear*, *happiness*, *disgust*, *anger*, *surprise*, *sad* and *like* are adopted. The corpus contains the emotion cauterization annotations for 10,000 Weibos.

Firstly, we select the Weibos with explicit emotion cause as the annotation target. Secondly, according to the relationship between “physiological arousal” and “expressive behaviors” from psychological review, both the emotion expression and emotion cause are annotated. Thirdly, according to the part of speech of the emotion cause, there are two major types of cause namely noun/noun phrase and verb/verb phrase. The examples for the noun/noun emotion causes are given below:

1. 我想,大概没有什么比世博会更能使上海人[自傲]的了。
*I think there is nothing more than **world expo** could make Shanghai citizens [feel proud] any more.*
2. 非常喜欢这片子的主题,人活着都是因为**有梦**。
*I [like] the **topic** of this film very much, every live for their dream.*

In the first example sentence, the cause of the emotion (bolded and underlined) is “世博会world expo” which is a noun phrase. For the second example, the cause of the “like” emotion is “主题topic”, which is a noun. Meanwhile, the emotion expressions are annotated with brackets.

The other kind of emotion cause is verb or verb phrase. Two examples sentences are given below.

3. 刚才打篮球赢了,[太激动了]。
*Just **win a basketball match**, it is [so exciting]!*
4. 从前台搜刮了一堆零食,[哈哈]。抱回自己办公室,这有点周扒皮的赶脚
***Plunder a lot of snack**, [LOL], get back to my office. It feels like Grandet.*

Here, the cause of emotion in this two sentences are “打篮球赢了win a basketball match” and “从前台搜刮了一堆零食plunder a lot of snack”, which are verb and verb phrase, respectively.

Up to now, we annotated the emotion expression and emotion cause in 1333 Weibos. In which, 722 (54.16% of all) emotion causes are nouns/noun phrases and 611 (45.83% of all) causes are verbs/verb phrases.

The observation on the annotated emotion cause corpus show that 796 causes are in the same sentence with the emotion expression. Meanwhile, 30.10% emotion causes occur before the emotion expression and only 9.49% emotion causes occur behind the emotion expression. The detail distribution information is listed in Table. 1.

Table 1. The distance from the cause to expression of emotion

Distance of cause	Number	Percent
In the same sentence	796	59.71%
Left 1 sentence	282	21.15%
Left 2 sentence	66	4.95%
Right 1 sentence	83	6.23%
Right 2 sentence	11	0.82%
Other	95	7.13%

The observation on the number of cause shows interesting results. Most Weibos (93.55% of all) have only one emotion cause while 82 Weibos have two emotion causes and 3 Weibos have four emotion causes. .

We also observe the distribution and characteristics of emotion expression in the corpus. Most emotions (1234 of 1333) are expressed by using the emotion words in Weibo while others use emotion icons.

The length of emotion words is also observed. The detail is shown in Table 2.

Based on the above observation and characteristics analysis, the methods for emotion cause detection are developed

Table 2. Distribution of the length of emotion words

Length	No.	Percent	Example
1	115	11.41%	One character word, such as “好 <i>good</i> ”
2	1033	78.08%	Two character word, such as “漂亮 <i>beautiful</i> ”
3	106	8.01%	Three character word, such as “够爷们 <i>real man</i> ”
4	32	2.41%	Four character word, such as “令人发指 <i>heinous</i> ”

3 Our Emotion Cause Detection Methods

Based on the observation on the annotated emotion cause corpus, we proposed three methods for identifying the sentence which contain the cause of emotion, namely emotion cause detection. One method is rule based and the other two are machine learning based.

The basic idea of our methods is to identify the cause candidate words (CCW) including nouns (noun phrase) and verbs (verb phrase) and determine whether they are cause of emotion. In the rule based method, we utilize the linguistic rules to decide if the CCW is a cause or not. Considering that from the point of view of psychological, the emotion expression should be helpful to emotion cause detection, we incorporate the linguistic clues and the emotion expression characteristics as features for the machine learning based method.

3.1 Rule-Based Emotion Detection

Rule-based method has shown efficient in emotion detection from news texts by Lee [10]. However, Weibo texts have characteristics different from the news texts. Thus, we construct an expanded rule set for emotion cause detection from Weibo text. Here, we firstly define the linguistic clues words for identifying emotion cause from the view of linguistics. Normally, the most linguistic cues words are verbs, conjunctions and prepositions. (Shown in Table 3)

Table 3. Linguistics clue words

Number	categories	Cue words
I	Independent conjunction	因为/because, 因/due to, 由于/because of, etc.
II	Coupled conjunction	,<(之)所以/so,(是)因为/because>,<(之)所以/so,(是)由于/due to> etc.
III	Preposition	为了/for, 以/according to, 因/due to,, etc.
IV	Verb of cause	让/make, 令/cause, 使/let
V	Verb of feeling	想到/think of, 谈起/speak of, 提到/mention, , etc.
VI	Others	的是/by the fact, 是/is, 就是/is, 的说 /said, etc.

Based on these linguistics clues, 18 rules are compiled to determine whether a CCW is a cause of an emotion expression. Here, Rule 1 - Rule 14 are the same as Lee’s work [10] . Rule 15- Rule 18 are new that are designed for emotion detection from Weibo texts. The detail of these rules is listed in Table 4.

Table 4. Expanded rules for emotion cause detection

No.	rules
	verb-object(C,E), E is verb
15	C=the sentence contains CCW and emotion word between which the dependency relation is verb-object
	E+“的/of”+C(F)
16	C=the focus sentence contains “de” and CCW
	C(contains “的/of”)+E
17	C= the focus sentence contains “de” and CCW
	C(B/F)+E
	E=special emotion expression in Weibo
18	C= the sentence contains special emotion word and CCW or the sentence before the focus sentence contains CCW

Here, *C* is the cause of emotion, *CCW* is the cause candidate word, *E* is the emotion expression word, *F* is the sub-sentence contains emotion expression word, *B* is the left sub-sentence of *F*, *A* is the right sub-sentence of *F*, *I*₁ is the former part of coupled conjunction and *I*₂ is the later part of coupled conjunction. For each *CCW* in all sub-sentences, we utilize the 18 rules to detect which one is the emotion cause.

3.2 Machine Learning Based Emotion Cause Detection

The above rule based method mainly uses linguistics features. Besides those features, the observation from the psychology view point show that the relation between emotion expression and its cause are also useful. Thus, the linguistics features and psychology based features are incorporated. Firstly, the mentioned 18 rules are converted to linguistic-based Boolean features. If a sub-sentence matches the rule, the value of corresponding feature is 1, otherwise 0. It is observed that the distance between *CCW* and emotion expression is helpful to determine whether *CCW* is the emotion cause. Thus, the distance is selected as a feature. If the *CCW* is in the same sentence with the expression, the value of distance feature is 0. If the sub-sentence of *CCW* is next to the emotion expression, the value of distance is 1, otherwise 2. Furthermore, considering that to the POS of emotion expression is helpful to determine the POS of emotion cause, all of the possible combination of POS patterns are also mapped as Boolean features. The complete feature space is shown in Table 5.

In this study, the SVMs and the CRFs are employed, respectively. For the SVMs based method, the emotion cause detection problem is transferred to a binary classification problem. For each sub-sentence, the SVMs classifier is employed to classify each sub-sentence to emotion cause or not. For the CRFs based method, the cause

detection problem is transferred to a sequence labeling problem. In this method, the Weibo is transferred to a sequence of sub-sentences. The CRFs is applied to label of each sub-sentence to 0-1 label. The 0 means the corresponding sub-sentence has no emotion cause and the 1 stands for the sub-sentence contains emotion cause.

Table 5. Feature space for machine learning based emotion detection

Feature categories	Description	Value
Linguistic feature	18 rules based on linguistic cues	0 or 1
Distance feature	Distance between cause and expression	0, 1, or 2
POS feature	All possible POS pattern combination of cause and expression	0 or 1

4 Experimental Results

The annotated emotion cause corpus on 1333 Weibos is adopted in this study to evaluate the proposed emotion detection method. The evaluation metric is the accuracy.

4.1 Evaluation on the Rule Based Method

In the first experiment, the proposed rule-based method (using 18 rules) are evaluated. Its performance is compared with Lee's rule based method which is regarded as reference baseline method. The achieved performance is listed in Table 6.

Table 6. The performances of rule-based methods

Method	Accuracy
Lee's rules (baseline)	40.12%
18 rules based method	68.30%

It is observed that our method improves 28.18% accuracy from Lee. It shows the effectiveness of the new proposed rules, especially the specific ones for Weibo text.

4.2 Evaluation on the Machine Learning Based Method

For the machine learning method, the 5-fold-validation is employed in the experiment. The achieved performances are shown in Table 7.

Table 7. The performance of machine learning based methods

Method	Accuracy
SVMs based	61.98%
CRFs based	77.57%

It is observed that the machine learning based methods further improve the accuracy of emotion detection. The majority of performance improvement attributes to the new features which considering the relationship between emotion expression and emotion cause. Furthermore, the CRFs based method considers the information between adjacent sentences and thus it achieves a better performance.

4.3 Further Analysis on CRFs for Emotion Cause Detection

According to 4.2, due to the sequence information, it is shown that CRFs achieves better performance. To further analyze the performance of CRFs-based method, the test samples are divided into sub-sentences. For the sub-sentences which contain emotion cause (EC) or no emotion cause (NEC). The achieved precision, recall and F-measure of EC and NEC are listed in Table 8, respectively.

Table 8. The performance of CRFs based method (sub-sentences)

Category	#correct	#total	#proposed	precision	recall	F-measure
EC	165	282	219	58.51%	75.34%	65.87%
NER	715	769	832	92.98%	85.94%	89.32%

It is observed that for the NEC sub-sentences, the CRFs method achieves a high precision and recall performance. The achieved F-measure for NEC sub-sentence classification is 89.32%. However, for the EC sub-sentences, the CRFs based method achieves a precision at 58.51% and 65.87% F-measure. These results mean that many NEC are wrongly classified as EC. It indicates that our feature space could cover the EC samples well, but it is not good enough to classify EC samples for NEC samples.

5 Conclusion

In this paper, an emotion cause corpus on Chinese Weibo text is designed and annotated. It is the first corpus for supporting the research of emotion cause detection from micro-blogging/Weibo text, based on our knowledge. Based on the observation on this corpus, three emotion cause detection methods, namely rule-based, SVMs based and CRFs based method are developed, respectively. The evaluations for these methods show that the CRFs based method achieves the best accuracy of 77.57% which is higher than the baseline method for 37.45%. The major improvement attributes to the new linguistics features which is specific to the Weibo text and the new features which considering the relation between emotion expression and emotion cause.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 61300112, 61370165, 61203378), Natural Science Foundation of Guangdong Province (No. S2012040007390, S2013010014475), MOE Specialized Research Fund for the Doctoral Program of Higher Education 20122302120070, Open Projects Program of National Laboratory of Pattern Recognition, Shenzhen

International Co-operation Research Funding GJHZ20120613110641217 and Baidu Collaborate Research Funding.

References

1. Mohammad, S.: From Once Upon a Time to Happily Ever After: Tracking Emotions in Novels and Fairy Tales. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pp. 105–114 (2011)
2. Purver, M., Battersby, S.: Experimenting with Distant Supervision for Emotion Classification. In: Proceedings of the 13th Conference of the EACL, pp. 482–491 (2012)
3. Panasenko, N., Trnka, A., Petranova, D., Magal, S.: Bilingual analysis of LOVE and HATRED emotional markers. In: Proceedings of the 3rd SAAIP workshop, IJCNLP 2013, Japan, pp. 15–23 (2013)
4. Vaassen, F., Daelemans, W.: Automatic Emotion Classification for Interpersonal Communication. In: Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (2011)
5. Tang, D., Qin, B., Liu, T., Li, Z.: Learning sentence representation for emotion classification on microblogs. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) NLPCC 2013. CCIS, vol. 400, pp. 212–223. Springer, Heidelberg (2013)
6. Xu, R., Ye, L., Xu, J.: Reader's Emotion Prediction Based on Weighted Latent Dirichlet Allocation and Multi-Label k-Nearest Neighbor Model. *Journal of Computational Information Systems* 9(6) (2013)
7. Yao, Y., Xu, R., Lu, Q., Liu, B., Xu, J., Zou, C., Yuan, L., Wang, S., Yao, L., He, Z.: Reader emotion prediction using concept and concept sequence features in news headlines. In: Gelbukh, A. (ed.) CICLing 2014, Part II. LNCS, vol. 8404, pp. 73–84. Springer, Heidelberg (2014)
8. Xu, R., Zou, C., Xu, J.: Reader's Emotion Prediction Based on Partitioned Latent Dirichlet Allocation Model. In: Proceedings of of International Conference on Internet Computing and Big Data (2013)
9. Ye, L., Xu, R., Xu, J.: Emotion Prediction of News Articles from Reader's Perspective based on Multi-label Classification. In: Proceedings of IEEE International Workshop on Web Information Processing, pp. 2019–2024 (2012)
10. Lee, S.Y.M., Chen, Y., Li, S., Huang, C.-R.: Emotion Cause Events: Corpus Construction and Analysis. In: Proceedings of International Conference on Language Resources and Evaluation (2010)
11. Chen, Y., Lee, S., Li, S., et al.: Emotion Cause Detection with Linguistic Constructions. In: Proceeding of International Conference on Computational Linguistics (2010)
12. Lee, S., Chen, Y., Huang, C., et al.: Detecting Emotion Causes with a Linguistic Rule-based Approach. In: Computational Intelligence (2012)
13. Das, D., Bandyopadhyay, S.: Analyzing Emotional Statements – Roles of General and Physiological Variables. In: Proceedings of IJCNLP (2011)
14. Myers, G.D.: *Theories of Emotion*. Psychology: Seventh Edition, p. 500. Worth Publishers, New York

News Topic Evolution Tracking by Incorporating Temporal Information

Jian Wang, Xianhui Liu*, Junli Wang, and Weidong Zhao

Engineering Research Centre of Ministry of Education
on Enterprise Digitalization Technology, Tongji University, China
lxh@tongji.edu.cn

Abstract. Time stamped texts or text sequences are ubiquitous in real life, such as news reports. Tracking the topic evolution of these texts has been an issue of considerable interest. Recent work has developed methods of tracking topic shifting over long time scales. However, most of these researches focus on a large corpus. Also, they only focus on the text itself and no attempt have been made to explore the temporal distribution of the corpus, which could provide meaningful and comprehensive clues for topic tracking. In this paper, we formally address this problem and put forward a novel method based on the topic model. We investigate the temporal distribution of news reports of a specific event and try to integrate this information with a topic model to enhance the performance of topic model. By focusing on a specific news event, we try to reveal more details about the event, such as, how many stages are there in the event, what aspect does each stage focus on, etc.

Keywords: Temporal Distribution, LDA, News Topic Evolution.

1 Introduction

With the dramatic increase of these digital document collections, the amount of information is far more beyond that person can efficiently and effectively process. There is a great demand for developing automatic text analysis models for analyzing these collections and organizing its contents. Probabilistic topic models, such as Latent Dirichlet Allocation (LDA) [1], Author-Topic Model [2] were proven to be very useful tools to address these issues.

With the need to model the time evolution of topics in large document collections, a family of probabilistic time series models were developed. Dynamic topic model (DTM) [3] captures the evolution of topics in a sequentially organized corpus of documents. Topic over Time (TOT) [4] model treats time as a continuous variable. Continuous Dynamic Topic Model (cDTM) [5] uses Brownian motion to model continuous time topic evolution. iDTM [6] is an infinite dynamic topic model which allows for an unbounded number of topics and captures the appearance and vanishing of topics. These models are quite useful when dealing with corpus with many different topics mixed up, but when it comes to a specific

* Corresponding author.

news event, the result seems to be not such remarkable. A couple of methods for generating timeline were proposed to deal with these issues in recent year [7], [8], [9].

However, most of these methods are trying to get a summarization of the event from the text but none of these methods mentioned above have taken the advantage of the temporal information as a prior knowledge. In this paper, we first explore the temporal distribution of news event, then propose an algorithm to automatically divide the corpus into different stages, in which the documents may have more coherence. By incorporating temporal information to topic model, we introduce a framework for tracking a specific news event evolution. The rest of this paper is organized as follows. In section 2, we illustrate the temporal distribution of the news event and describe the division algorithm in detail. In section 3, we propose our analysis framework and explain how temporal information can enhance the topic model. In section 4, we present the case study experiment in detail. In section 5, we conclude the paper with some analysis and outlook for future work.

2 Temporal Distribution of News Events

When a sensational event burst out, related reports will overflow in media very soon. Later the quantity of related reports would gradually decline. But once new details are disclosed or someone else is involved, the event gains its popularity again and likewise the amount of related reports would move up sharply. Suppose we label each of the popular period as a stage. Generally, most news events may have several stages, and each stage has its own focus. The results showed in the Fig.1 are exactly in conformity with what we've assumed above.

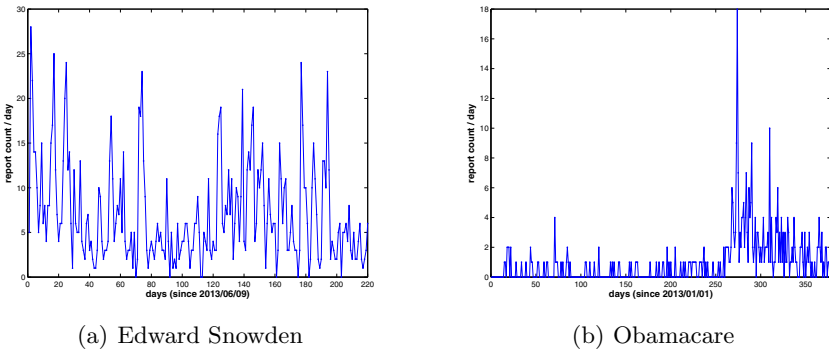


Fig. 1. Temporal distribution of news reports quantity about "Edward Snowden" and "Obamacare". X-axis represents day's interval from the beginning date and the Y-axis represents article count. Articles were crawled from The Guardian with the key words "Edward Snowden" from 9 June 2013 to 10 Jan 2014, and "Obamacare" from 1 Jan 2013 to 10 Jan 2014.

2.1 Documents Division and the Adaptive K-Means Algorithm

In Blei's DTM [3], the corpus was evenly divided by time, thus every episode has the same time scale. Let's take a look at the Fig.1, if the documents are divided by time evenly, the dividing points may just locate at the peak point. Intuitively, this is not a good choice. Because the reports around the peak point mainly focus on the same aspect and they have a strong coherence.

So, we propose a simple but efficient method which is called the Adaptive K-Means algorithm. This algorithm is based on the K-Means algorithm[10] and could automatically divide the documents without setting the cluster number K in advance. In this paper the cluster number K means the episode number. The algorithm is described as follow:

Algorithm 1: Adaptive K-Means algorithm

Data: X : news count of each day; max_k : the maximum k ; t : threshold value

Result: $count$: article count of each episode; $dists$: weighted mean distance array; K : the best count of cluster

```

 $Y \leftarrow$  remove zero points from  $X$ 
for  $i \leftarrow 1$  to  $max\_k$  do
    [ $count, sumd$ ] =  $kmeans(Y, i)$ ;
    //  $count$ : point count of each cluster
    //  $sumd$ : sum distance of each cluster
     $means \leftarrow calc\_mean\_distance(count, sumd)$ ;
    //  $means$ : mean distances of all clusters
     $dists[i] \leftarrow calc\_weighted\_mean\_distance(means)$ ;
    if  $i > 1$  then
        if  $dists[i] - dists[i - 1] < t$  then
            |  $K \leftarrow (i - 1)$ ; break;
        end
    end
end
end
if  $K = 0$  then
    |  $K \leftarrow max\_k$ ;
end

```

The Adaptive K-Means algorithm starts with a small number of clusters, and adds the number one by one. At each iteration of the algorithm, we calculate the *weighted mean distance* of all clusters. The *weighted mean distance* is defined as follows:

$$Weighted\ Mean\ Distance = \frac{\sum_{i=1}^n \text{mean distance of cluster } i}{n} \quad (1)$$

The distance calculated in the Equation.1 refers to Euclidean distance. In the beginning, the number of centers is much smaller than the best K , so the *Weighted Mean Distance* would decline rapidly. With the number getting closer to the best

K , the decrease value becomes smaller and smaller. Once the decrease value is smaller than a specific threshold value, then the current number of centers is regarded as the best K .

3 Incorporating Temporal Information into Topic Model

In this section, we illustrate how to use topic model to track news topic evolution and why temporal information can improve the analysis result.

3.1 Basic Concepts

First of all, we would like to give the definitions of some basic concepts which would be frequently mentioned.

1. A **stage** is a time episode in which documents have a strong coherence, and documents are likely related to a same aspect of the event
2. The **main topic** could run throughout all stages and is the line connecting all the episodes;
3. The **auxiliary topics** are all the other topics besides the main topic, which present the new aspects of the main topic in different stages; The auxiliary topics could be regarded as the progresses of the event because they very a lot along the time

In order to discover the main topic and track the evolution, we need to calculate the similarity between adjacent episodes. As the topic is characterized by a distribution over words. A simple measure method of similarity between topics is the *Kullback-Leibler divergence* (also called *relative entropy*). However, the *KL divergence* is not a proper distance measure method because it is not symmetric. An alternative option is the *Jensen-Shannon distance*, which is a smoothed and symmetric extension of the *KL divergence*. For discrete probability distribution P and Q , the *JS Distance* is defined to be

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \quad (2)$$

With the averaged variable $M = \frac{1}{2}(P + Q)$.

3.2 Framework of Our Method

Our analytical framework is based on the LDA [1], a generative latent variable model that treats documents as bags of words generated by one or more topics. We perform parameter estimation using collapsed Gibbs sampling [11] [12]. We could firstly divide the corpus into several subsets by time, and apply LDA within each subset, respectively. As for the division, we've described the *Adaptive K-Means algorithm* above which makes more sense than the method of simply dividing the corpus by time evenly

By incorporating temporal information, the overall framework of analysis process is as follows:

1. Prepare documents for each episode with *Adaptive K-Means algorithm*;
2. Preprocess of the documents in each episode;
3. Draw topic distribution of each episode from topic model (LDA);
4. Discover the main topic and draw the evolution map of the event.

3.3 Document Coherence and Evaluation

Document Coherence measures the topic similarity among documents. Intuitively, if articles within a corpus are more coherent, more detail could be revealed by topic models. In order to better present document coherence we propose a new evaluation method which is called *n Topic Coverage Rate*(TCR_n).

$$TCR_n = \frac{\|articles\ belong\ to\ these\ n\ topics\|}{\|all\ articles\|} * 100\% \quad (3)$$

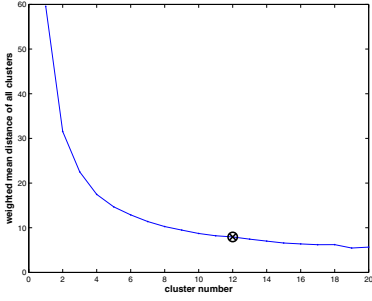
In this equation, $\|\cdot\|$ denotes the element count of a collection. TCR_n measures the documents cover rate of the top n topics within the whole corpus. From the Equation.3, we can see that with topic number n fixed, the bigger the TCR_n is, the more coherent the articles are. In other words, with the TCR_n fixed, the smaller the n is, the more coherent the articles are.

4 Experiment Result and Analysis

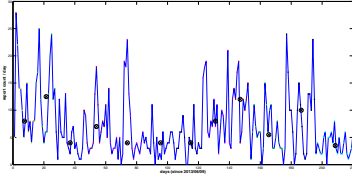
In this section we illustrate the result of the topic model with temporal information incorporated. First of all, we demonstrate the division result of the *Adaptive K-Means algorithm* with two corpora, and later focus on one of them to give a deep illustration. We analyze 1550 documents crawled from the Guardian with the key words "Edward Snowden" which is one of the top events of 2013. The time of these documents varies from June 9 of year 2013 to the end of year 2013. Our corpus is made up of approximately 1.5 million words. First of all, we use Stanford Parser¹ to parse the full text, and only keep words which are noun, proper noun, verb and adjective. Next, we lemmatize the remaining words. At last, we prune the vocabulary by removing stop words and removing terms that occurred less than 5 times. The remaining vocabulary size is 7732.

Fig.2 shows the result after applying the *Adaptive K-Means algorithm*. Our corpus of "Edward Snowden" is divided into 12 subsets. For the sake of document coherence comparison, we also divide the corpus into another 12 subsets by time evenly. We set the initialization parameters as follow: LDA parameters, $\alpha=2$ and $\beta=0.5$; Gibbs sampling parameters, total iterations=1000, burn-in iterations=200, sample interval=3. After running topic model in each episode, respectively, we calculate TCR_n of all episodes. Fig.3(a) is an example of top 5 topics' coverage. Obviously, our division algorithm has a higher coverage. To be more convincible, we calculate the average coverage rates of all episodes with different topic numbers, and the Fig.3(b) shows that our method has a general advantage.

¹ <http://nlp.stanford.edu/software/lex-parser.shtml>



(a) Edward Snowden



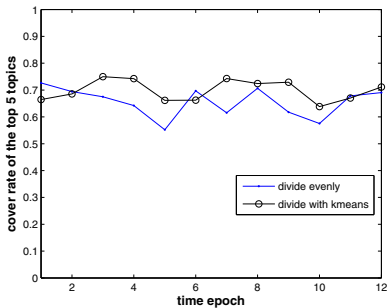
(b) Edward Snowden

Fig. 2. Cluster number determination process and documents division results by applying adaptive K-Means algorithm. The black \otimes indicates the best number of clusters in (a), and the centre point in (b). In (b), data points in different clusters are labelled with different colours. (the maximum value of episode is 20 and the threshold value is 0.5)

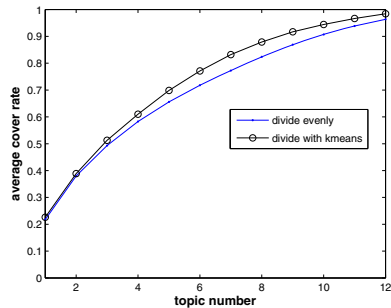
4.1 Evolution Map of "Edward Snowden"

Considering that the corpus is about one specific event and there won't be too many aspects in each episode, so we only pick the top 3 topics of each episode to draw the **evolution map**.

Fig.4(a) is the evolution map of the event "Edward Snowden" drawn from the method introduced in this paper. On this map, the main topic which runs throughout all stages is the one chained with arrow lines. To make a contrast, we also apply the DTM to generate evolution map Fig.4(b). From the Fig.4(b), we can see that topics in all the episodes seem to be almost the same. That's because DTM assumes that topic number is fixed during all episodes and no new topic would emerge and all the topics are evolved from the first episode. Thus, it

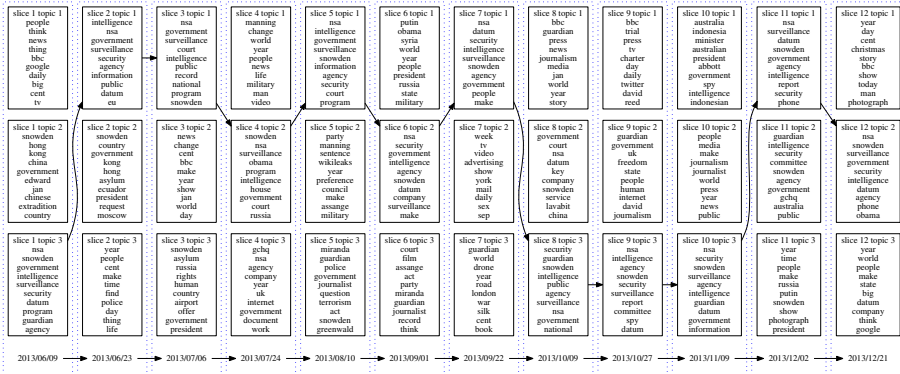


(a) TCR_5

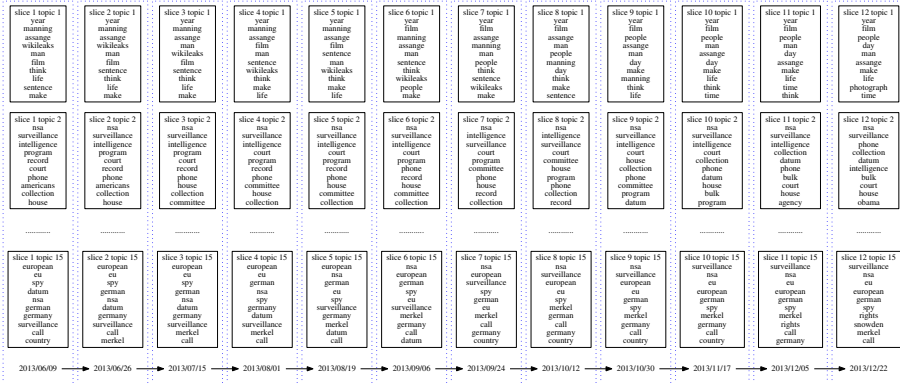


(b) Average TCR_n

Fig. 3. TCR of corpus about "Edward Snowden"



(a) Topics evolution map of the event "Edward Snowden" by the method of this paper



(b) Topics evolution map of the event "Edward Snowden" by the DTM

Fig. 4. Evolution Map

is suitable for the corpus that contains many different topics, such as academic documents. While in this paper, we concentrate on a specific news event, so it is not applicable.

5 Conclusion

In this paper, we address the problem of modeling sequence documents related to a specified news event. We explore the temporal distribution of news reports and treat them as a prior knowledge of the sequence topic model. By incorporating the temporal information we can generate an evolution map of a specific event. In the future, we plan to model the entities involved in the event and explore how they influence the event evolution. We also intend to develop an interactive system to better explore the event detail.

Acknowledgement. This work is supported by the National Science Foundation of China under grant No. 61105047. This work is also partially supported by the Research Program of Science and Technology Commission of Shanghai Municipality of China under grant No 12dz1125400 and 13111103100. Here, we would like to express our deep gratitude.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)
2. Michal, R.-Z., Thomas, G., Mark, S., Padhraic, S.: The author-topic model for authors and documents. In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 487–494. AUAI Press (2004)
3. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: *Proceedings of the 23rd International Conference on Machine learning, ICML 2006*, pp. 113–120. ACM, New York (2006)
4. Wang, X., McCallum, A.: Topics over time: A non-Markov continuous-time model of topical trends. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 424–433. ACM (2006)
5. Chong, W., David, B., David, H.: Continuous Time Dynamic Topic Models. In: *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI) (2008)*
6. Ahmed, A., Xing, E.P.: Timeline: A dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. *arXiv preprint arXiv:1203.3463* (2012)
7. Tang, S., Zhang, Y., Wang, H., Chen, M., Wu, F., Zhuang, Y.: The discovery of burst topic and its intermittent evolution in our real world. *Communications, China* 10(3), 1–12 (2013)
8. Zehnalova, S., Horak, Z., Kudelka, M., Snasel, V.: Evolution of Author's Topic in Authorship Network. In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pp. 1207–1210. IEEE Computer Society (2012)
9. Lin, C., Lin, C., Li, J., Wang, D., Chen, Y., Li, T.: Generating event storylines from microblogs. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 175–184. ACM (2012)
10. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(7), 881–892 (2002)
11. Griffiths, T.L., Steyvers, M.: Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101(suppl. 1), 5228–5235 (2004)
12. Heinrich, G.: Parameter estimation for text analysis (2005)

Author Index

- Chen, Guang 160
Chen, Long 181
Chen, Qingcai 150
Chen, Yijiang 46
Chen, Ying 299
Chen, Zhumin 229
- Dalielihan, Kamali 103
Diao, Yigang 392
Du, Jinhua 91
- Feng, Jian 67
Feng, Shi 238, 286
Feng, Yansong 333, 392
- Gao, Ji-xun 422
Gao, Sheng-xiang 422
Gui, Lin 457
Guo, Jun 160
- Hao, Hongwei 79, 357
He, Tingting 138
He, Yan 150
He, Zhicheng 168
Hong, Xu-dong 422
Hong, Yu 414
Hu, Baotian 150
Huang, Degen 181
Huang, Yalou 168
- Jiang, Tingsong 23
Jin, Qin 217
- Kang, Longbiao 150
Kong, Fang 13
- Li, Chengxin 217
Li, Dong 168
Li, Guangyi 403
Li, Juanzi 251
Li, Junjie 205
Li, Peng 67
Li, Shoushan 430
Li, Xiao 103
Li, Xuwei 379
- Li, Zhaohui 168
Li, Zhixing 251
Lin, Lei 368
Lin, Liyuan 430
Liu, Bin 457
Liu, Bingquan 67, 368
Liu, Chenglin 79
Liu, Chunyang 205
Liu, Feng 67
Liu, Jiangming 123
Liu, Jie 168
Liu, Kaiyu 311
Liu, Kehui 379
Liu, Lu 54
Liu, Mengting 238
Liu, Ming 67
Liu, Qun 113
Liu, Ting 193
Liu, Xianhui 465
Liu, Yiqun 263
Long, Wen-xu 422
Lu, Huanquan 392
Lu, Qin 457
Lu, Xiaoqing 54
Luo, Bingfeng 392
Lv, Xueqiang 379
- Ma, Jun 229
Ma, Shaoping 263
Mao, Jiaxin 263
Meng, Zeyu 449
Mi, Chenggang 103
- Pang, Lin 205
Pei, Bei 299
- Qi, Zhenyu 357
Qin, Bing 193
Qu, Jingwei 54
- Ren, Pengjie 229
- Sha, Lei 23
Su, Chang 46
Sui, Xueqin 229
Sui, Zhifang 23

- Sun, Chengjie 368
 Sun, Hong 321
 Sun, Yaming 368

 Tang, Zhi 54
 Tian, Jia 46

 Wang, Daling 238, 286
 Wang, Hao 357
 Wang, Houfeng 345, 403
 Wang, Huiyuan 275
 Wang, Jian 465
 Wang, Junli 465
 Wang, Lei 103
 Wang, Miaomiao 91
 Wang, Mingqiang 238
 Wang, Peng 79
 Wang, Xiaolong 67
 Wang, Xing 414
 Wang, Yongtao 54
 Wang, Yuan 168
 Wang, Zhongqing 430
 Wei, Furu 321
 Wen, Siqiang 251
 Wen, Wushao 438
 Wu, Huimin 217
 Wu, Kai 229
 Wu, Xiangping 150

 Xia, Yunqing 275, 311
 Xie, Jun 123
 Xie, Zhongda 275
 Xiong, Deyi 414
 Xu, Bo 79, 357
 Xu, JinAn 123
 Xu, Kun 333
 Xu, Ruifeng 457
 Xu, Weiran 160
 Xun, Endong 449

 Yang, Haitong 1
 Yang, Ping 438
 Yang, Sitong 113
 Yang, Yating 103
 Yao, Jianmin 414
 Ye, Chengxu 438
 Yu, Dong 449
 Yu, Heng 113
 Yu, Zheng-tao 422
 Yuan, Li 457

 Zhang, Heng 79
 Zhang, Le 286
 Zhang, Meng 91
 Zhang, Min 263, 414
 Zhang, Qiuge 275
 Zhang, Sheng 333
 Zhang, Weitai 160
 Zhang, Xiaodong 345
 Zhang, Yifei 238, 286
 Zhang, Yujie 123
 Zhao, Dongyan 333, 392
 Zhao, Huan 275, 311
 Zhao, Jun 138
 Zhao, Weidong 465
 Zhao, Yanyan 193
 Zhou, Fengyu 229
 Zhou, Guangyou 138
 Zhou, Guodong 13, 34, 430
 Zhou, Huiwei 181
 Zhou, Ming 321
 Zhou, Yu 205, 457
 Zhu, Hualing 311
 Zhu, Qiaoming 34
 Zong, Chengqing 1
 Zou, Bowei 34
 Zou, Xianqi 368